

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SIMULACE DATOVÝCH PŘENOSŮ V INTEGROVANÝCH SÍTÍCH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL ROZSYPAL

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SIMULACE DATOVÝCH PŘENOSŮ V INTEGROVANÝCH SÍTÍCH

SIMULATION OF DATA TRANSMISSIONS IN INTEGRATED NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL ROZSYPAL

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. VLADISLAV ŠKORPIL, CSc.

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Pavel Rozsypal

ID: 134395

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Simulace datových přenosů v integrovaných sítích

POKYNY PRO VYPRACOVÁNÍ:

Rozeberte problematiku datových přenosů v integrovaných sítích se zaměřením na kvalitu služby QoS. Analyzujte možnosti využití nástrojů Matlab a Simulink při simulaci činnosti různých síťových prvků (např. switch, router, hub, bridge), kodeků, neuronových sítí apod. Výstupy analýzy aplikujte při návrhu počítačových cvičení zaměřených na telekomunikační služby a komunikační systémy obecně. V případě zjištění vhodnosti simulace také pomocí OPNET IT Guru popřípadě jiného simulačního software ji lze do počítačových cvičení rovněž zařadit. Simulace navrhnete tak, aby bylo umožněno porovnání různých alternativ (např. způsoby řízení síťového prvku s uvažováním kvality služby). Porovnání realizujte, popište výhody a nevýhody jednotlivých alternativ. Výstupem práce musí být zadání minimálně tří počítačových cvičení včetně vzorových protokolů.

DOPORUČENÁ LITERATURA:

- [1] CHAO, H.J., GUO, X. Quality of Service Control in High-Speed Networks. John Wiley, New York 2002
- [2] BLUNÁR, K., DIVIŠ, Z. Telekomunikační sítě. VŠB-TU, Ostrava 2003.
- [3] PUŽMANOVÁ, R. PUŽMANOVÁ, R. Moderní komunikační sítě A-Z. Computer Press, Brno 2007

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem této práce bylo rozebrat problematiku datových přenosů v integrovaných sítích se zaměřením na kvalitu služby. V práci jsou popsány různé síťové prvky, jejich funkce, výhody a nevýhody. Pozornost byla věnována také problematice neuronových sítí se zaměřením na analogii mezi biologickým neuronem a umělým neuronem. Dále se práce zabývala možnostmi programu MATLAB pro simulace těchto sítí. Pro simulaci chování síťových prvků byl použit program OPNET IT Guru.

KLÍČOVÁ SLOVA

Kvalita služby, QoS, integrovaná síť, síťový prvek, neuronové sítě, MATLAB, OPNET IT Guru

ABSTRACT

The goal of this assignment was to analyze the issue of data transfer in integrated networks focusing on quality of service. The thesis describes the various network elements, their function, advantages and disadvantages. Attention was also paid to neural networks, with a focus on the analogy between biological neuron and the artificial neuron. Another part of paper was dedicated to the simulation possibilities of these network in MATLAB. To simulate the behavior of network elements was used OPNET IT Guru software.

KEYWORDS

Quality of service, QoS, integrated network, network element, neural network, MATLAB, OPNET IT Guru

ROZSYPAL, Pavel *Simulace datových přenosů v integrovaných sítích*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 65 s. Vedoucí práce byl doc. Ing. Vladislav Škorpil, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Simulace datových přenosů v integrovaných sítích“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce panu doc. Ing. Vladislavu Škorpilovi, CSc. za odbornou pomoc a cenné připomínky při zpracování této práce.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
(podpis autora)

OBSAH

Úvod	10
1 Teoretický rozbor problematiky	11
1.1 Integrované sítě	11
1.2 Kvalita služby QoS	11
1.2.1 Integrované služby (IntServ)	13
1.2.2 Diferencované služby (DiffServ)	18
1.3 Aktivní síťové prvky	20
1.3.1 Hub-Rozbočovač	20
1.3.2 Router-Směrovač	20
1.3.3 Bridge-Most	23
1.3.4 Switch-Přepínač	24
1.4 Neuronové sítě	27
1.4.1 Biologický neuron	27
1.4.2 Umělé neuronové sítě	28
1.4.3 Umělý neuron	28
1.4.4 Formální neuron	29
1.4.5 Přenosové funkce [4]	30
1.5 Kodeky	33
1.5.1 Příklady kodeků	33
2 Využití programů pro simulace daných problematik	34
2.1 Matlab	34
2.1.1 Neural Network Toolbox	34
2.2 OPNET IT Guru Academic Edition	36
3 Závěr	41
Literatura	43
Seznam symbolů, veličin a zkratk	44
Seznam příloh	45
A Příloha 1	46
A.1 Základy k neuronovým sítím	46
B Příloha 2	50
B.1 Konfigurace QoS mechanismu OPNET IT Guru	50

C Příloha 3	59
C.1 Síťové prvky	59

SEZNAM OBRÁZKŮ

1.1	Základní části implementované v IntServ	13
1.2	Základní operace protokolu RSVP. [6]	17
1.3	Referenční model technologie IntServ. [6]	18
1.4	Příklad rozbočovače: Cisco-Linksys EFAH08W EtherFast 8-port hub.	20
1.5	Příklad routeru: TP-LINK TL-WR1043ND Ultimate Wireless N Gigabit Router.	22
1.6	Příklad mostu: Cisco WAP2000, Wifi Access Point.	24
1.7	Příklad přepínače: Cisco Catalyst 2950 24 Switch.	26
1.8	Ukázka biologického neuronu	28
1.9	Základní model jednoduchého neuronu McCulloch-Pittsovův perceptron [4]	29
1.10	Heavisideova přenosová funkce	30
1.11	Dvouhodnotová přenosová funkce	31
1.12	Přenosová funkce - sigmoida	31
1.13	Přenosová funkce - s omezením	32
1.14	Lineární přenosová funkce	32
2.1	Grafické rozhraní pro tvorbu neuronových sítí	34
2.2	Úvodní plocha programu OPNET IT Guru	36
2.3	Ukázka <i>Helpu</i> pro předpřipravený projekt IP_QoS	37
2.4	Ukázka editoru	38
2.5	Hlavní menu projektového editoru a hlavní ikony projektového editoru	39
2.6	Paleta objektů – knihovna komponent pro ethernetové sítě	39
B.1	Nastavení FIFO	51
B.2	Graf zobrazující množství zahozených paketů pro jednotlivé mechanismy	55
B.3	Graf zobrazující zpoždění paketů služby Voice pro jednotlivé mechanismy	55
B.4	Graf zobrazující koncové zpoždění paketů pro síť s a bez podpory QoS	57
B.5	Graf zobrazující kolísání koncového zpoždění paketů pro síť s a bez podpory QoS	57
C.1	Celkové schema zapojení	64
C.2	Subsystem síťový prvek	64
C.3	Subsystem adresace	65
C.4	Subsystem priority	65

ÚVOD

V dnešní době používá telefony, počítače a podobná zařízení většina populace. V počátcích vývoje komunikačních sítí bylo jejich používání známo pouze v malé míře. Požadavky na přenos dat v síti byly dlouho na základní úrovni a nebylo potřeba jakkoliv řešit různé priority pro různé typy dat. Velký pokrok v počítačových a komunikačních technologiích však přinesl postupně se zvyšující nároky na přenos dat v síti a základní třída pro přenos dat „best effort“ přestala být postačující. Bylo nutné vyvinout nové mechanismy pro klasifikace a řízení přenosu dat v sítích. Tyto mechanismy mají za úkol zajistit jistou kvalitu pro přenos dat v síti, tedy pomáhají zajistit, aby námi vyslaný paket opravdu a v pořádku doputoval sítí do svého cíle.

V práci byla přiblížena problematika přenosů dat v dnešních sítích a základní představa o QoS a jejich modelech IntServ a DiffServ. Dále byly rozebrány funkce, výhody a nevýhody nejčastěji používaných aktivních prvků sítí. Dále se práce zabývala základními poznatky o neuronových sítích, které lze využít k řízení funkcí těchto prvků. Nakonec byl přiblížen použitý program OPNET IT Guru pro simulace činnosti aktivních prvků a možnosti využití MATLABu pro simulace neuronových sítí. V každém z těchto programů bylo vytvořeno zadání pro počítačová cvičení, jež mají studentům přiblížit danou problematiku.

1 TEORETICKÝ ROZBOR PROBLEMATIKY

1.1 Integrované sítě

Když se začínalo s vývojem komunikačních sítí, existovaly pro všechny služby jednotné požadavky na přenos dat. Jelikož bylo málo síťových služeb a využívání přenosových linek bylo malé, nebyl důvod řešit nějaké úrovně kvality pro poskytované služby. Internet pracuje jako paketově orientovaná síť, kde je každý z paketů přenášen samostatně přes celou síť. Z tohoto důvodu nebyl nijak garantován čas doručení paketu od stanice, která ho vyslala, k cílové stanici. Daný paket mohl být také na kterémkoliv z mnoha síťových uzlů z různých důvodů zahozen.

Tím jak se rychle rozvíjely komunikační technologie a jak se změnily síťové služby, přišly od jejich uživatelů vyšší nároky na jejich přenos. Největší nároky jsou kladeny na služby pracující v reálném čase. S vyšším množstvím služeb přichází více požadavků na jejich parametry přenosu po síti.

Internet nebyl na tento vývoj připraven a bylo nutné vytvořit dodatečné mechanismy, které podporují různé zacházení při přenosu s různými typy dat.

Důraz je kladen především na schopnosti sítě při rozlišování datových toků a pak jejich rozdělování na jednotlivé třídy provozu. Po rozboru údajů o vytvořených třídách lze určit politiku provozu tak, aby co nejvíce zaručovala dodržení požadavků na přenos dat v síti.[1][2][3]

1.2 Kvalita služby QoS

V internetu se pro přenos a spojování IP paketů používá především metoda „Best Effort“. Při této metodě se síť snaží přenést každý paket co nejrychlejším a nejefektivnějším způsobem ke svému cíli.

Některé typy provozu vedle dat v síti (např. video v reálném čase nebo hlas) jsou náchylné na odchylky ve zpoždění a na ztrátu paketů. Každý druh provozu také potřebuje jinou šířku pásma. Proto různé kategorie provozu potřebují garanci různých veličin od sítě ve formě kvality služby (QoS, Quality of Service). Při vývoji IP se předpokládaly různé stupně QoS, které však nebyly využívány.[1]

Hodnotit QoS nelze jedinou hodnotou, protože její konkrétní vyjádření by bylo složité a v hojně většině případů nepřesné.

QoS lze klasifikovat na základě různých měřitelných parametrů síťového provozu, např.:

- **Zpoždění paketů** je doba potřebná pro přenos paketu od zdroje k jeho cíli.
- **Ztrátovost paketů** udává množství paketů, které nedorazí do cíle. Tato hodnota je často udávána v procentech (poměr počtu ztracených k celkovému počtu odeslaných paketů).
- **Šířka pásma** je přenosová kapacita, která souvisí s propustností (objem úspěšně přenesených dat za jednotku času)
- **Kolísání zpoždění paketů – Jitter** značí rozdíly v intervalech mezi přicházejícími pakety.

Protože různé síťové služby jsou odlišně citlivé na parametry provozu, je nutné stanovit základní třídy provozu. Správné zařazení služby do třídy provozu je klíčové pro výslednou kvalitu služby. Rozlišujeme tedy:

- Klasický přenos souborů (FTP, pošta, záznamy databází), které jsou:
 - citlivé na ztrátu a chybný příjem paketů,
 - necitlivé na zpoždění a jitter.
- Služby pracující v reálném čase (audio a video aplikace), které jsou
 - vysoce citlivé na zpoždění a jitter,
 - méně citlivé na ztrátu paketu.

Základní úkoly QoS mechanismů:

- Třídění datových jednotek
 - Klasifikace paketů
 - Značkování paketů
- Odlišné zacházení pro jednotlivé třídy
 - Dohled nad provozem
 - Aktivní správa front
 - Plánované odesílání paketů
 - Tvarování provozu

QoS architektura v Internetu

Při vývoji IP se předpokládaly různé stupně QoS, které však nebyly prakticky využívány. V posledních letech se využívají modely které jsou nazývány model Integrovaných služeb (IntServ) a model Diferencovaných služeb (DiffServ). Modely jsou navrhnuté tak, aby byly schopny samostatné činnosti. V reálném provedení však většinou nepracují samy, ale vzájemně se podporují.

1.2.1 Integrované služby (IntServ)

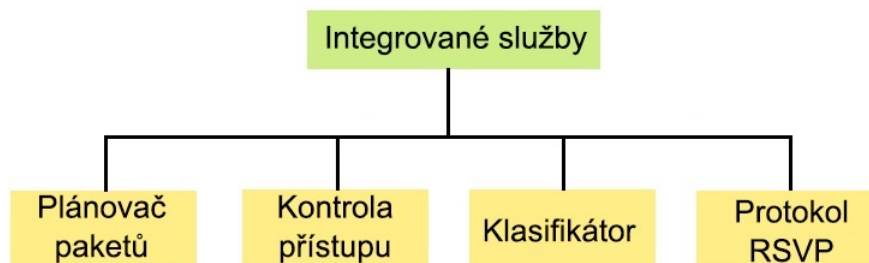
Architektura IntServ byla první, jež zaručovala určitou kvalitu služeb. Definována byla v roce 1994 v dokumentu RFC 2212.

Vývoj IS architektury je vázán na dvě výkonné technologie:

- IP multicast: důležitý aspekt pro multimediální komunikaci
- služby v reálném čase: audio a video aplikace, které vyžadují přísnou kontrolu zpoždění paketů konec-konec

Garance pro služby v reálném čase předpokládají použití IS směrovačů s možností rezervování prostředků sítě z důvodu poskytování na aplikaci závislé kvality služby. Rezervování síťových prostředků pro vybraný tok bitů také předpokládá dohled nad stavem každého datového toku v každém IS směrovači, což je podstatný rozdíl od původního internetu. „Best Effort“ princip zabezpečuje stejné zpracování všech paketů. „Best Effort“ směrovač nashromáždí všechny IP pakety pro určitý výstup do jedné řady typu „First in - First out“. Naproti tomu IS architektura garantuje QoS jednotlivých datových toků. [1]

IntServ architektura obsahuje 4 základní části:



Obr. 1.1: Základní části implementované v IntServ

Plánovač paketů

Do směrovače přichází na vstupní porty pakety různou rychlostí, s různou kapacitou a různým obsahem. Každý paket je pak směrován na určený výstupní port, odkud pokračuje dále k cíli. Jelikož v daný okamžik může z daného portu odejít pouze jeden paket, jsou další pakety řazeny do fronty. Je potřeba rozhodnout, který z přicházejících paketů by měl následovat za paketem, který právě opouští směrovač. Plánovač paketů je tedy určen pro plánování paketů ve frontách. Při pevné velikosti šířky pásma výstupního portu je nutné, aby byly vstupní toky spravedlivě rozloženy. Plánování paketů je základ mechanismů QoS.

Základní typy plánování paketů

- FIFO (First in first out)
- PQ (Priority Queuing)
- WFQ (Weighted Fair Queuing)

FIFO je nejjednodušší technika řazení paketů do fronty výstupu ze směrovače. Zařazování probíhá na principu „kdo dřív přijde“, takže směrovač neřeší žádné priority datových toků na vstupu.

Výhody:

- rychlost v malých sítích, kde neprobíhají velké datové toky

Nevýhody:

- nerozlišuje typy síťového provozu
- není možné poskytovat různé stupně QoS
- při zahlcení sítě mají všechny pakety stejnou šanci na to, že budou zahozeny

PQ technika je založena na sestavení několika front typu FIFO s tím, že do každé této fronty jsou posílány pakety dané priority. Fronty s vyšší prioritou mají vždy přednost před těmi s nižší prioritou. To znamená, že pokud probíhá datový tok s nejvyšší prioritou, tak pouze pakety z této fronty mohou jít na výstup směrovače. Pakety v nižší prioritní frontě tedy čekají, dokud neskončí provoz na vyšších prioritách.

Výhody:

- pakety datových toků s vyšší prioritou nebudou zahazovány
- tato technika už zajišťuje jistou úroveň QoS

Nevýhody:

- každá nižší fronta nemůže poslat na výstup pakety, pokud je v jakékoliv vyšší frontě provoz
- v sítích, kde je velké množství prioritního provozu dochází k potlačení ostatních provozů (tento stav byl nazván jako „monopolizace“ určitého síťového provozu)

WFQ vzniklo rozšířením techniky FQ (Fair Queuing, metoda spravedlivého řízení front). Pomocí této metody byl vyřešen problém „monopolizace“, který vznikl u PQ. Ve směrovači je opět definováno několik prioritních front. Každá tato fronta má svou váhu, která určuje, jakou část dostupné šířky pásma může provoz v dané frontě zabrat. Při odesílání paketů je použita cyklická obsluha front, takže se nestává, že by byl zastaven provoz na frontách s nižší prioritou.

Výhody:

- na výstup směrovače jde provoz ze všech front
- dochází k malému zahazování paketů i v nižších frontách (množství je závislé na velikosti provozu v síti)

Nevýhody:

- složitější řízení odesílání paketů
- větší zátěž pro směrovač

Kontrola přístupu

Aby architektura mohla fungovat jiným způsobem než je služba „best effort“, je nutný blok nesoucí název **kontrola přístupu**. Tento blok zajišťuje službu, která na žádost aplikace o poskytnutí síťových prostředků (vyrovnávací paměť, šířka pásma) pošle pozitivní nebo negativní odpověď. Bez tohoto bloku by byly poskytovány všechny dostupné zdroje všem třídám paketů bez rozdílu.

Klasifikátor paketů

Hlavní funkcí klasifikátoru paketů je rozhodování, zda daný paket patří k datovému toku s garantovaným zacházením, a pokud ano, tak ke kterému. Každý paket je přiřazen k datovému toku s určitou garancí zacházení nebo v případě, že žádné nevyhovuje, zůstává v datovém toku, který je pracován metodou „Best-effort“. Klasifikace pracuje nejčastěji na základě zdrojové a cílové IP adresy, identifikátoru protokolu vyšší vrstvy a čísla portů transportního protokolu. Ve chvíli, kdy je paket

zařazen do některého z rezervovaných toků, je paket předán do bloku plánovače paketů. [6]

Protokol RSVP(Resource reSerVation Protocol)[7]

Pro informování síťových prvků, a tak i k rezervaci síťových prostředků, využívá model IntServ tzv. rezervační protokoly. Nejčastěji využívaným protokolem je RSVP (Resource reSerVation Protocol), který byl definován v RFC 2205. Tento protokol poskytuje mechanismus pro vytváření a správu rezervačního stavu po celé cestě mezi zdrojem datového toku a jeho cílem. Rozhodnutí, která trasa bude pro přenos dat vybrána ale stále náleží směrovacímu protokolu. RSVP proces se řídí informacemi ze směrovacích tabulek a na jejich základě posílá RSVP zprávy. [6].

Protokol RSVP definuje 2 základní typy zpráv:

- PATH zpráva
- RESV zpráva

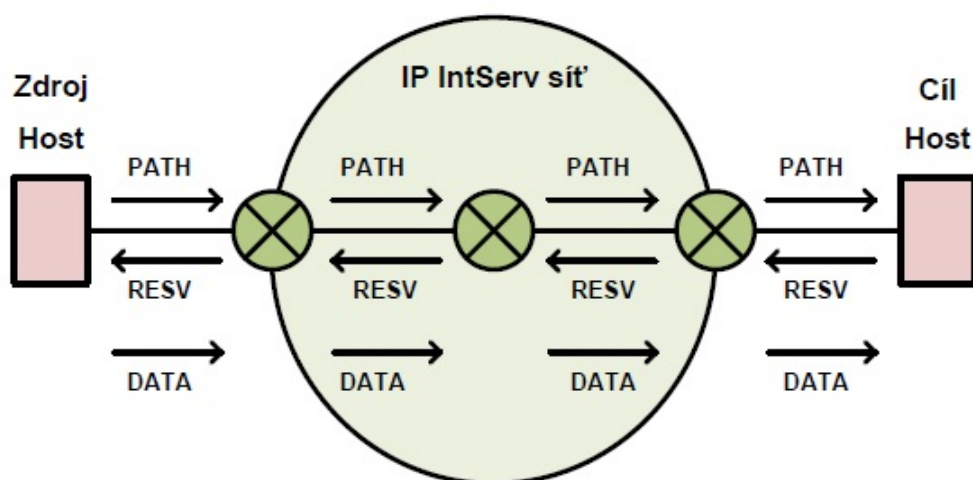
Zdroj datového toku (aplikace, působící jako odesílatel) posílá v pravidelných intervalech zprávy typu PATH. Tato zpráva prochází každým směrovačem v cestě k cíli a obsahuje záznam o typu vysílané informace. Hlavně je však do ní zapisována cesta mezi zdrojem a příjemcem dat. Tyto informace pak slouží při rezervaci síťových prostředků. Když příjemce obdrží zprávu PATH, tak na ni odpovídá zprávou RESV, kterou potvrzuje přidělení síťových prostředků. Zpráva RESV je vysílána cestou popsanou v přijaté zprávě PATH.

Zpráva PATH obsahuje:

- Previous Hop
- Sender template
- Sender TSpec
- Sender AdSpec

Zpráva RESV obsahuje:

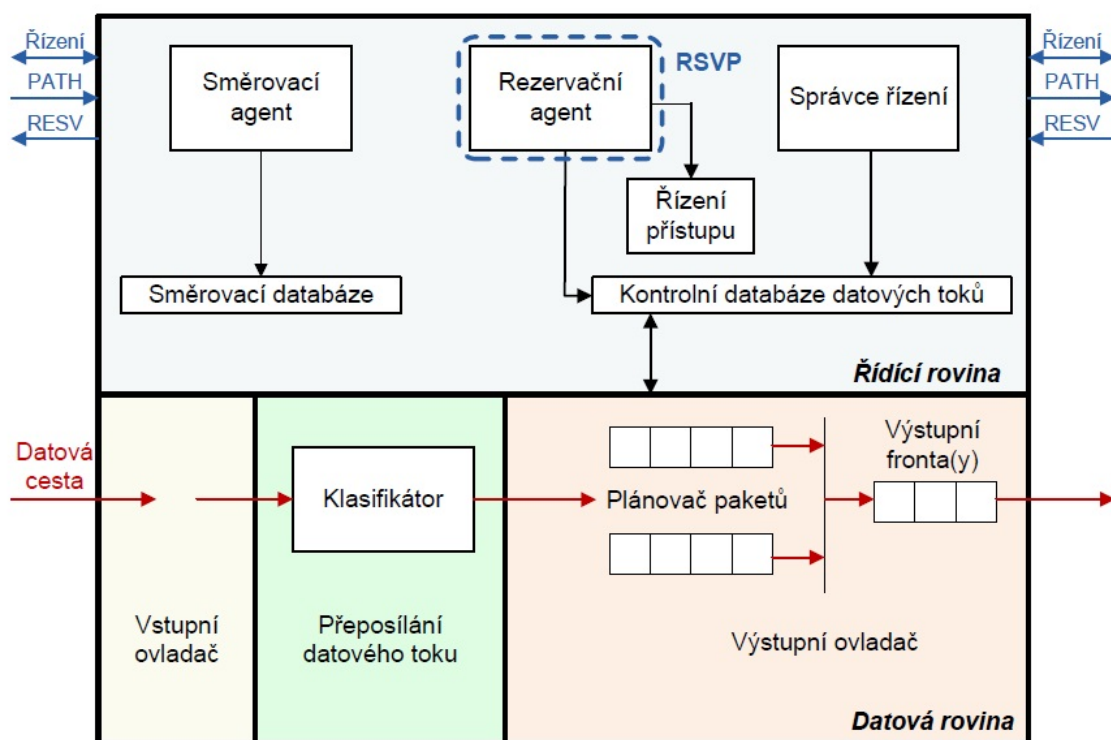
- Flow Spec
- Filter Spec



Obr. 1.2: Základní operace protokolu RSVP. [6]

Stav rezervace vytvořený pomocí RSVP má definovanou dobu trvání a po jejím uplynutí tento stav automaticky končí. Proto je standardně každých 30 sekund tento stav aktualizován pomocí RSVP zpráv. Díky tomu lze také reagovat na změny v topologii sítě (výpadek směrovače, přerušení vedení). [6]

RSVP rezervuje pouze v jednom směru, proto je v případě obousměrné komunikace nutné provést rezervaci pro každý směr zvlášť. [6]



Obr. 1.3: Referenční model technologie IntServ. [6]

1.2.2 Diferencované služby (DiffServ)

Největší problém DiffServ (Differentiated Service) architektury je v rozlišování sítí a přenosových rychlostí na rozhraních. DiffServ architektura je silně spojově orientovaná. [2]

Vlastnosti:

- udržuje všechny důležité informace o individuálním datovém toku až po konec sítě,
- realizuje na konci sítě všechny komplexní akce vztahující se k jednomu datovému toku. Aby se celá komplexnost realizovala na konci sítě, vychází uvedená zásada z toho, že QoS signalizace musí být explicitně přenášena v záhlaví každého IP datagramu. Pokud se QoS definuje jen na základě informace v záhlaví, soustřeďuje to datový provoz do omezeného počtu kategorií. Za tímto účelem bylo v paketu definováno tzv. DS pole pro diferencování služeb.

Mechanismus diferencovaných služeb využívá bity DSCP. Byly specifikovány tři základní kategorie chování v rámci skoku (PHB, Per Hop Behavior), tedy mezi dvěma sousedními uzly: [2]

- **urychlené předávání** (EF, Expedited Forwarding) - požadované chování pro daný typ služby (např. u VoIP minimální ztráty, latence). EF nabízí absolutní záruky kolísání zpoždění pro třídu provozu, proto je velmi složité na zajištění a neefektivní, protože poskytnutí EF danému toku odpovídá poskytnutí virtuálního okruhu, což vede k malému využití síťových prostředků. Z toho důvodu lze poskytovat EF pouze omezenému počtu toků.
- **zajištěné předávání** (AF, Assured Forwarding) - služba modelovaná podle služby CIR u Frame Relay má zajistit přenos datagramů sítí garantovanou rychlostí. Využívá protokol TCP a pracuje na základě stanovení priorit pro různé kategorie provozu. AF specifikuje čtyři nezávislé třídy pro přesný přenos. Každá třída má přiděleny prostředky pro přenos datagramů (šířku pásma, vyrovnávací paměť). Mimo to jsou definovány 3 úrovně priorit zničení.
- **základní služba** (BE, Best Effort) - vhodná především pro datové přenosy.

1.3 Aktivní síťové prvky

1.3.1 Hub-Rozbočovač

Rozbočovač je aktivní síťový prvek, který umožňuje její větvení a je základním kamenem pro sítě s hvězdicovou topologií. Chová se jako opakovač. Funkcí rozbočovače je jednoduché kopírování dat přivedených na jeden z portů na všechny ostatní porty, a to bez zjišťování, na který port daná data náleží. To má za následek, že všechna zařízení v síti „vidí“ všechna síťová data, a to zapříčiňuje přetěžování segmentů, kterým data nejsou určena. Proto jsou rozbočovače vhodné jen pro malé sítě se slabším provozem. Ve větších sítích byly přepínače nahrazeny dokonalejšími zařízeními, a to směrovači.



Obr. 1.4: Příklad rozbočovače: Cisco-Linksys EFAH08W EtherFast 8-port hub.

1.3.2 Router-Směrovač

Základní funkce směrovače: [2]

- zjištění a výběr „nejlepší“ cesty v síti,
- vnitřní přepínání paketů ze vstupního portu na výstupní.

Směrovače jsou zařízení pracující na síťové vrstvě a mají tedy větší možnosti uplatnění než jiné komponenty. Jejich vnitřní inteligence umožňuje rozhodnutí „směrovat, či nesměrovat“. Směrovače pracují se síťovými adresami, fyzické adresy jim nic neříkají. Směrovač zajímá jen ta část cílové adresy paketu, která označuje umístění zařízení v konkrétní síti. V důsledku toho obsahuje směrovací tabulka jen záznamy o dostupných sítích či podsítích a o odpovídajících portech směrovače, které se používají k přesměrování paketu a metriku (délku cesty) dané cesty. Záznamy o všech připojených stanicích směrovací tabulka neobsahuje. Směrovací tabulka je založena

na logickém uspořádání sítě, tzn. že při výběru cesty se směrovač řídí nejdelší hodnotou prefixu souhlasícího s cílovou IP adresou směrovaného datagramu.

Směrovač se také chová jinak k paketům s všeobecnou adresou nebo s adresou, která nemá záznam ve směrovací tabulce. Tyto pakety směrovač nepřesměrovává na všechny porty, ale buď tyto pakety ničí nebo je přesměruje na tzv. implicitní síť (síť pro neidentifikované adresy). Jako taková může být zvolena např. síť propojující podnikovou síť s větší sítí, ve které je směrování záležitostí vnitřních směrovačů a nezajímá směrovače v malé síti. [2]

Směrovač při příjmu paketu zkoumá kromě cílové adresy také informace v jiných polích. Jednou z funkcí směrovače je číst a měnit pole v paketu známé jako doba životnosti (time to live). Každý paket dostává maximální možnou hodnotu počtu směrovačů, ve kterých může být zpracováván. Každý směrovač, který daný paket zpracovává tuto hodnotu sníží o 1. V případě, že je hodnota doby životnosti na vstupu směrovače 0 a není v síti s koncovým zařízením, ničí směrovač daný paket a vygeneruje chybovou hlášku. Tím se zamezí nekonečnému bloudění paketu se špatným směrováním v síti.

Směrovače také musí být schopny provést fragmentaci paketů tam, kde má přijatý paket vyšší metriku, než je povolená MTU na výstupním portu. Při fragmentaci musí směrovač všechny potřebné informace pro opětovné sestavení zaznamenat ve všech fragmentech paketu.

Směrovače spolu musí komunikovat, předávat si informace o aktuální situaci cest v síti a musí být schopny informovat zdrojovou stanici o vzniklých problémech při posílání datagramu dále v síti.

Hlavním úkolem směrovačů a směrovacích protokolů je rychlý proces konvergence, což je stav, při kterém směrovače v síti vidí její topologii stejně a směřují správně podle aktuálního stavu sítě. Ztrátou konvergence se rozumí např. výpadek jednoho ze směrovačů. V tom případě musí směrovače reagovat a poslat příslušné informace o novém stavu sítě a aktualizovat si své směrovací tabulky. [2]

Výhody směrovačů [2]

- segmentace sítí s cílem zamezit průchod chybným paketům, paketům s všeobecnou nebo neznámou cílovou adresou,
- zvýšení bezpečnosti a průchodnosti v síti pomocí inteligentní filtrace paketů na základě síťových adres a typů protokolů a na základě filtrace směrovacích informací,
- zvýšení průchodnosti sítě pomocí inteligence pro podporu frontových mechanismů (místo jedné fronty lze konfigurovat více front podle priority paketů)
- možnost používání linky pouze v případě skutečné potřeby, využívání záložního vytáčeného spojení,
- podpora rozkládání zátěže do paralelních cest,
- možná náhrada páteřní sítě výkonným směrovačem se sdruženou vnitřní páteřní sběrnici,
- podle typu výrobce určitá míra podpory automatické konfigurace směrovače, konfigurace na dálku prostřednictvím serveru.

Nevýhody směrovačů [2]

- nároky na konfiguraci
- nároky na správu a management v síti



Obr. 1.5: Příklad routeru: TP-LINK TL-WR1043ND Ultimate Wireless N Gigabit Router.

1.3.3 Bridge-Most

Základní vlastnosti mostů:

- přepínají a filtrují rámce,
- rozhodují se na základě fyzických adres příchozích rámců a svých adresových tabulek,
- kontrolují rámce z hlediska chyb a specifické řízení toku,
- vykonávají úkoly vyplývající z příslušných metod přístupu (např. CSMA/CD),
- podporují multiprotokolové prostředí

Mosty byly prvními zařízeními, která skutečně propojovala sítě a podsítě. Původně se mosty orientovaly na přímé propojování lokálních sítí, ale později se normalizovala i jejich možná podpora propojování lokálních sítí na dálku, a to prostřednictvím přenosového média jiného typu.

Mosty umožňují snížení provozu v síti, protože propouštějí jen rámce, které jsou určeny stanicím v jiné části sítě. Když jsou rámce s adresou cílové stanice na straně přijímacího portu, jsou udržovány lokálně, protože je most zastaví před dalším postupem sítí. Takle filtrace je automatická u všech mostů, navíc lze manuálně konfigurovat podporu pro filtraci na základě informací v rámcích, např. MAC adres a typu síťového protokolu. Rámce s všeobecnou nebo neznámou individuální adresou však most propouští. [2][1]

Mosty pracují formou uložit a pošli. Nejprve musí celý rámec přijmout a teprve pak ho zpracovávají na základě cílové MAC adresy pro rozhodnutí o přepnutí mezi porty (popř. na základě dalších polí použitých pro filtraci). Tento režim práce umožňuje mostům kontrolovat rámce z hlediska chyb, takže nepropustí chybné rámce (např. příliš krátké nebo dlouhé).

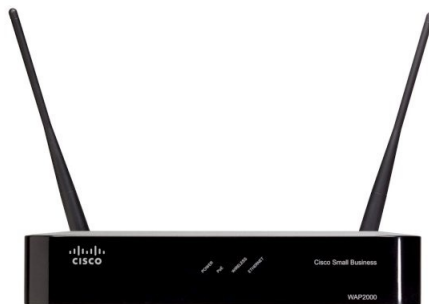
Mosty jsou závislé na typu sítí, které propojují. Nejčastěji se jedná o propojení homogenní, tedy propojení stejných lokálních segmentů. Ve složitějším případě vzniká heterogenní síť.

Výhody komunikačních mostů [2]

- snižují velikost kolizních domén
- rozšiřují síť, které dosáhly svého fyzického limitu (např. délky segmentu u Ethernetu)
- vytváří přechod mezi sítěmi s různou přenosovou rychlostí
- jsou schopny propojit segmenty sítě s protokoly nepodporujícími směrování
- propojení síťových segmentů s různými síťovými architekturami
- nižší cena
- není třeba konfigurovat

Nevýhody komunikačních mostů [2]

- podpora jediné domény pro rámce na všeobecnou adresu
- neschopnost fragmentace rámců
- závislost na médiu (typu lokální sítě a metody přístupu k médiu)
- omezená schopnost filtrace



Obr. 1.6: Příklad mostu: Cisco WAP2000, Wifi Access Point.

1.3.4 Switch-Přepínač

Přepínače mohou být definovány jako mosty s více porty. S mosty mají společné zásadní vlastnosti vyplývající z vrstvy protokolové architektury, na které oba pracují.

Podobně jako mosty se i přepínače učí znát MAC adresy připojených zařízení za porty, propojují podobné lokální sítě, nemění rámce při zpracování a filtrují na základě MAC. Význam a obsah rozhodovací tabulky je stejný jako u tabulky transparentních mostů. Tabulka adres přiřazených portům je dynamická. Při každém průchodu rámce s cílovou MAC adresou se tato adresa v tabulce aktualizuje. Při delší neaktivitě se daná adresa z tabulky vymaže. V případě rámce s neznámou cílovou nebo všeobecnou adresou se synchronně zkopírují na všechny výstupní porty s výjimkou vstupního portu rámce.[2]

Přepínače umožňují vyhrazené připojení, podporují virtuální lokální sítě a nabízejí nám výběr režimu práce (ulož-pošli, průběžné zpracování). Jejich práce je rychlejší (rozhodování a přepínání je hardwarová záležitost, u mostů to byl software) a typicky se nepřipojují k rozsáhlým sítím. Přepínače se uplatňují ve stejných lokálních sítích, které se odlišují maximálně rychlostmi na jednotlivých portech.

Příkladem vhodného použití přepínače je řešení zahlcení sdílené sítě serverů a klientů. Přepínání jednotlivých skupin na jednotlivé servery zprůchodní síť a umožní hladší komunikaci s jednotlivými servery. Přepínač ale nevyřeší situaci, kde je jediný exponovaný server, se kterým všechny stanice nejčastěji komunikují. Zahlčení v tomto případě způsobí nedostatečný paměťový a procesní zdroj - jeden server. Řešením by bylo přidání dalších serverů a přechod ze sdílené na přepínanou síť.[2]

V síti by mělo být dodrženo pravidlo fyzické a logické blízkosti: [2]

- Nejčastěji používaný server by měl být fyzicky i logicky nejblíže uživateli nebo pracovní skupině, tzn. na stejném segmentu sdílené nebo přepínané sítě, ne za směrovačem,
- síťové prostředky jako tiskárny by měly být rovněž v blízkosti pracovních skupin, v případě velkého objemu dat pro tisk by měly být dokonce na vyhrazeném segmentu připojeném prostřednictvím přepínače.

Přepínače dovoluují seskupovat stanice na svých portech (segmentované lokální sítě) nebo připojit na port jedinou stanici a vytvořit tím soukromou lokální síť.

Výhody přepínačů [2]

- vyšší výkon a snížení zpoždění v síti
- jednodušší správa a management sítě
- nižší náklady (oproti směrovačům)
- segmentace sítě až do přidělení šířky pásma v jedné stanici
- podpora virtuálních sítí – segmentace do domén (VLAN) pro všeobecně adresované rámce

Nevýhody přepínačů [2]

- podpora jediné domény pro rámce určené na všeobecnou adresu (v rámci přepínané sítě bez VLAN)



Obr. 1.7: Příklad přepínače: Cisco Catalyst 2950 24 Switch.

1.4 Neuronové sítě

1.4.1 Biologický neuron

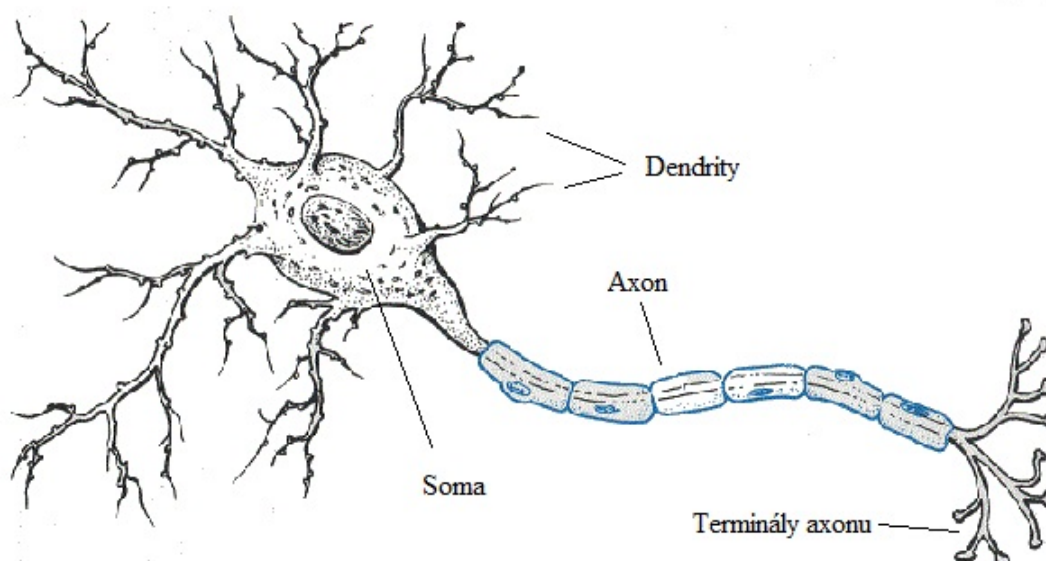
Neuron neboli nervová buňka je základním stavebním prvkem nervové soustavy každého živého organismu. Jedná se o samostatnou živou buňku, která má co nejúčelněji zpracovávat, uchovávat a přenášet informace. Byla popsána řada druhů neuronů u živých organismů, avšak každý z nich má stejnou základní strukturu:

- **soma**
- **axon**
- **terminály axonu**
- **dendrity se synapsemy**

Neuron tedy tvoří základní tělo s buněčným jádrem **soma**. Z tohoto základního těla vychází jediný, i když rozvětvený výstup zvaný **axon**. Axon je z vnějšku a po téměř celé své délce obalen ochranou vrstvou, která by se dala popsat podobně jako izolace kabelu, která se nazývá myelinová pochva. Tato izolace axonu je na určitých částech přerušena. Těmito místům se říká Ranvierovy zářezy. Každý axon je ukončen řadou větví nazvaných **terminály axonu**. Další částí neuronu jsou výběžky zvané **dendrity**. Dendrity jsou prakticky vstupy do buňky.[4]

Aby mezi sebou mohly neurony přenášet informace, musí být propojeny. To nám zajišťují právě výše uvedené terminály axonů a dendrity. Terminály axonů jsou zakončeny blankou, která se dotýká výběžků dendritů jiných neuronů. Výběžky dendritů jsou tzv. trny. Místo, kde se axon dotýká dendritu jiného neuronu, je tedy místo, kde dochází k předávání informací mezi jednotlivými neurony. Tohle místo nazýváme **synapse**. Synapse se z funkčního hlediska dělí na **excitační** a **inhibiční**. Excitační synapse v nervové soustavě budí a rozšiřují vzruch, naopak inhibiční jej utlumují. [4]

Pro přenos informace je zásadní význam dán membránovému potenciálu. Ten je výsledkem činnosti sodíko-draslíkové pumpy, která zajišťuje přenos iontů sodíku ven a iontů draslíku do buňky touto membránou. Na této membráně vzniká potenciálový rozdíl. Neuronové buňky mají schopnost podráždění. Můžou reagovat na elektrický, chemický nebo mechanický podnět (záleží na druhu synapse). Může se jednat o podráždění místní (to se dále nešíří) nebo může dojít ke vzniku akčního potenciálu. Vznik akčního potenciálu je odpovědí na nadprahový podnět. Důsledkem toho je, že změny potenciálu způsobí vzruch, který nezůstává na místě, ale začíná postupovat axonem. Tento vruch se šíří konstantní rychlostí a s konstantní amplitudou. Při podprahovém podnětu dochází k lokální reakci, která velmi rychle zaniká.[4]



Obr. 1.8: Ukázka biologického neuronu

1.4.2 Umělé neuronové sítě

Obecně je umělá neuronová síť struktura pro zpracovávání dat, která je obvykle složená z vysokého počtu vzájemně propojených prvků (neuronů). Každý může přijímat libovolný počet vstupních dat, pak může na libovolný počet dalších neuronů předávat informace o svém stavu. Umělé neuronové sítě se dnes využívají k řešení praktických problémů jako je rozpoznávání písma, převod mluvené řeči na text, rozpoznávání obrazu, identifikace podpisů a aj.

1.4.3 Umělý neuron

Umělý neuron je velmi podobný tomu biologickému, ale neplatí, že se chovají stejně. Skládá se ze základních částí stejně jako biologický. Umělý neuron má n vstupů, které slouží k propojení s ostatními neurony v síti. Každý ze vstupů má svou váhu. Váhy spojení ovlivňují jednotlivé vstupy do neuronů a tím celou neuronovou síť. Hodnota váhy nám udává, jak moc ovlivní daný vstup výstup z neuronu. Tělo umělého neuronu zpracovává informace ze vstupů dle zadané funkce a výsledek je posílán na výstup neuronu.

1.4.4 Formální neuron

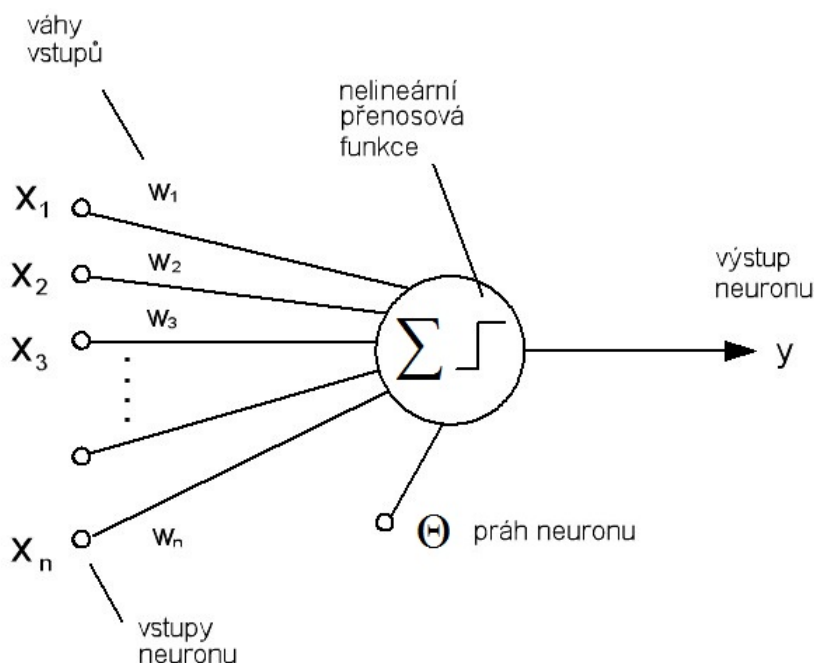
Matematické modely neuronu jsou základními kameny umělých neuronových sítí. Existuje mnoho matematických modelů neuronů. Liší se svou topologií a především matematickou funkcí.

Formální neuron je jeden z nejznámějších modelů umělých neuronů. Tento model je nazýván také McCulloch-Pittsův perceptron podle svých autorů. [4]

Vstupní údaje jsou zpracovávány dle vztahu:

$$y = S(\sum_b^a w_i x_i + \Theta), kde \quad (1.1)$$

- y je výstup neuronu,
- x_i je i -tý vstup neuronu, maximální počet vstupů je n ,
- w_i představuje hodnotu i -té synaptické váhy,
- S je nelineární přenosová funkce neuronu,
- Θ představuje prahovou hodnotu.



Obr. 1.9: Základní model jednoduchého neuronu McCulloch-Pittsův perceptron [4]

1.4.5 Přenosové funkce [4]

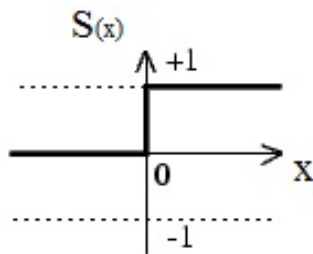
Přenosové funkce jsou také označovány jako **aktivační** funkce. Tyto funkce slouží k převodu vnitřního potenciálu do definovaného oboru výstupních hodnot. V oblasti neuronových sítí se nejčastěji využívají 4 funkce:

- Heavisideova (také prahová),
- dvouhodnotová,
- sigmoida.
- funkce s omezením.

Pro **prahovou** funkci platí:

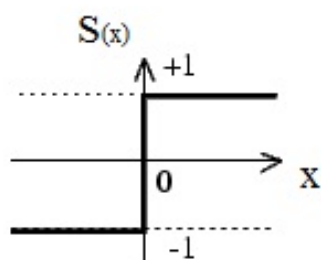
$$S(x) = \begin{cases} 0, & \text{pro } x < 0 \\ 1, & \text{pro } x \geq 0 \end{cases} \quad (1.2)$$

Výstup neuronu u této funkce nabývá pouze hodnot 0 nebo 1. Hodnoty 0 nabývá, pokud má vstup hodnotu menší než 0. Hodnoty 1 nabývá výstup v případě, kdy vstup je rovný nebo větší jak 0.



Obr. 1.10: Heavisideova přenosová funkce

V některých případech je ale vhodnější, aby byl obor hodnot výstupu $\{1, -1\}$. V takovýchto případech se využívá **dvouhodnotová** funkce. Tato funkce je hodně podobná Heavisideově funkci, ale platí, že když chceme na výstupu neuronu hodnotu 1, tak musí být vstupní hodnota větší než prahová hodnota této funkce a naopak pro hodnotu -1 musí být na vstupu hodnota menší, než je prahová hodnota této funkce.

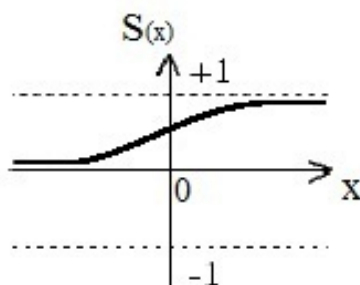


Obr. 1.11: Dvouhodnotová přenosová funkce

Sigmoida je asi nejčastěji používanou přenosovou funkcí. Jejím příkladem je:

$$S(x) = \frac{1}{1 + e^{-ax}}. \quad (1.3)$$

Hodnoty sigmoidy se v mínus nekonečnu blíží nule a v nekonečnu k jedničce. Pro nulu je to hodnota 0,5. Výhodou této funkce je existence derivace v každém bodě. Proto se používá u sítí se spojitými vstupy a výstupy (např. v sítích typu back-propagation).

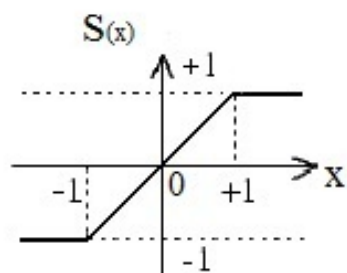


Obr. 1.12: Přenosová funkce - sigmoida

Funkce **s omezením** je jakousi alternativou hyperbolické tangenty (ta rozšiřuje obor výstupních hodnot sigmoidy na $\{-1,1\}$). Je to taková funkce, kde není kladen požadavek na diferencovatelnost aktivační funkce.

$$S(x) = \begin{cases} -1, & \text{pro } x \leq -1 \\ x, & \text{pro } -1 \leq x \leq 1 \\ 1, & \text{pro } 1 \leq x \end{cases}$$

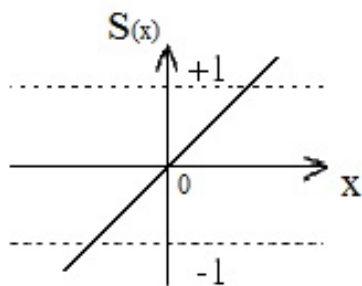
(1.4)



Obr. 1.13: Přenosová funkce - s omezením

Další funkcí je například funkce **lineární**, kde na výstupu dostáváme hodnotu vstupu.

$$S(x) = x \quad (1.5)$$



Obr. 1.14: Lineární přenosová funkce

1.5 Kodeky

Slovo kodek vzniklo složením KOdér a DEKodér. Jak už název napovídá, jedná se o fyzická zařízení či programy, které jsou schopny daným způsobem změnit datový proud. Kodeky data mění na zakódovaný signál, který je například vhodnější pro přenos po síti nebo využívány pro zašifrování daného obsahu.

Velmi často jsou kodeky využívány při videokonferencích, streamování videa apod. Tato multimédia obsahují části, které každé náleží jinému hardwaru, programu či procesu. Pro přenos je ale vhodné je svázat do jednoho celku.

1.5.1 Příklady kodeků

- Zvukové kodeky
 1. WavPack
 2. Musepack
 3. Speex
 4. Windows media Audio
- Video kodeky
 1. DivX Pro Codec
 2. Xvid
 3. Cinepack
 4. Windows media Video
 5. Huffyuv kodek

2 VYUŽITÍ PROGRAMŮ PRO SIMULACE DATNÝCH PROBLEMATIK

2.1 Matlab

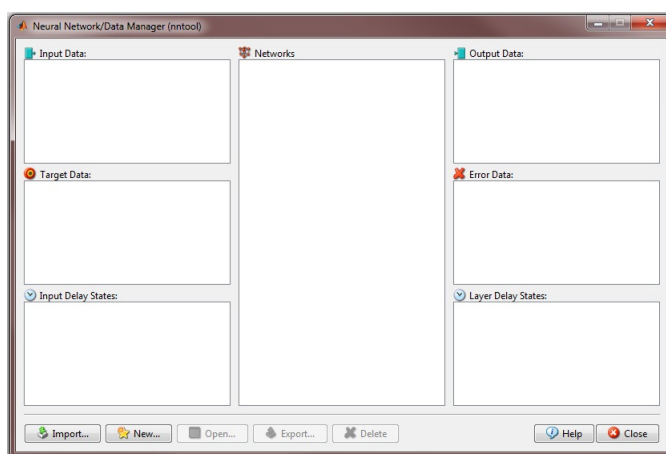
MATLAB (od společnosti MathWorks) byl původně určen pro matematické účely, ale časem do něj byly přidány nové funkce a rozšíření. MATLAB umožňuje počítání s maticemi, vykreslování 2D a 3D grafů funkcí, implementaci algoritmu, počítačové simulace, analýzu a prezentaci dat. Základní vlastností je, že veškeré objekty jsou považovány za prvky v maticích. Od toho byl i odvozen název - MATrix LABoratory.

MATLAB a jeho výkonnost se rozšiřují díky přidávání nových a nových souborů (toolboxů) nebo uživateli sestavených programů v tzv. m-files. Toolboxy se zaměřují na řešení či simulace daných problému, pro které byly navrženy.

2.1.1 Neural Network Toolbox

Pro práci s umělými neuronovými sítěmi byla vytvořena knihovna (Neural Network Toolbox), která umožňuje uživatelům MATLABu jednoduše pracovat s různými neuronovými sítěmi. S její pomocí lze definovat různé struktury neuronových sítí (např. vícevrstvé perceptronové sítě) a simulovat jejich výpočty.

Pro rychlé a jednoduché simulace neuronových sítí slouží grafické rozhraní spustitelné v příkazovém řádku MATLABu pomocí příkazu `nntool`.



Obr. 2.1: Grafické rozhraní pro tvorbu neuronových sítí

V tomto prostředí lze rychle importovat, vytvářet, simulovat a hodnotit výkon neuronové sítě. Prostředí nabízí širokou škálu nastavování od typu sítě (např. dopřednou neuronovou síť, Hopfieldovu síť, perceptron), přes trénovací, učící funkce, až po počty vrstev sítě a počty neuronů. Další možností tohoto rozhraní je zobrazení vytvořené sítě, její trénování a také simulování. Výsledky simulace se ukládají do výstupních dat - Output Data. Užitečnou možností je určitě i exportování vytvořené a odsimulované sítě do Workspace MATLABu.

Při použití grafického rozhraní však nejsme schopni ovlivnit řadu věcí. Pro plnohodnotnou práci s neuronovými sítěmi je vhodné pracovat v základním rozhraní MATLABu. Zde lze určit, z jakých souborů budeme načítat potřebná data, kde se budou ukládat výsledky a sledovat, jak se program chová.

V programu MATLAB byla vypracována dvě počítačová cvičení (viz A a C) včetně zadání, postupu vypracování, kontrolních úkolů a požadovaných výsledků pro učitele.

2.2 OPNET IT Guru Academic Edition

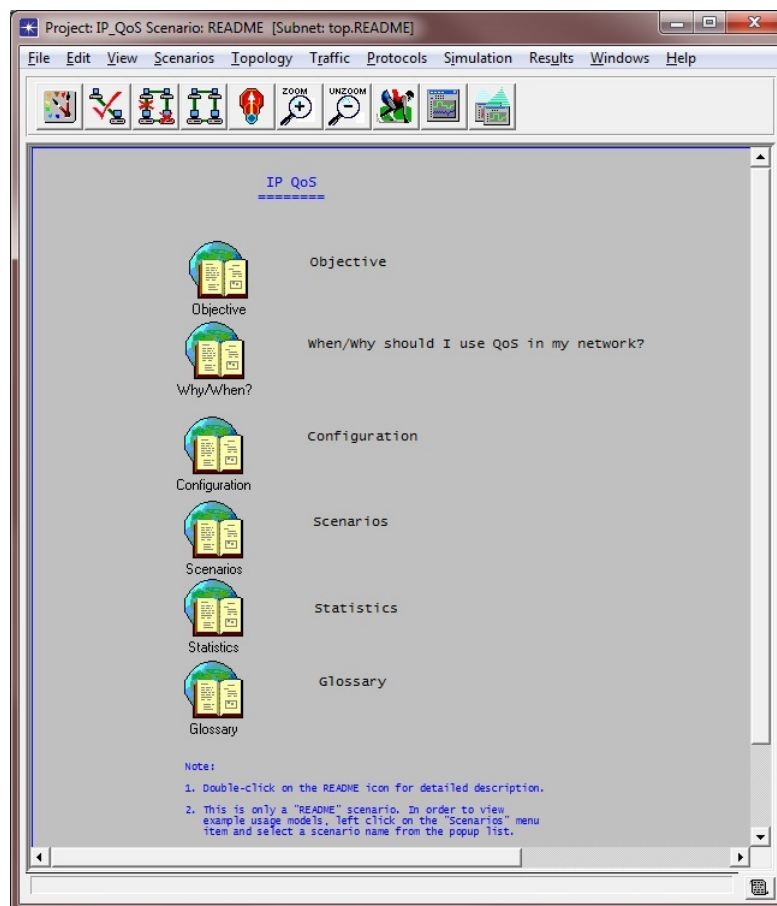
Simulace chování a funkce aktivních síťových prvků pomocí programu MATLAB je náročná. Pro návrh, simulaci a rozbor výsledků se mnohem více hodí program Opnet IT Guru od společnosti OPNET Technologies. V tomto programu lze modelovat i velké sítě s možností přesně nastavit požadované vlastnosti. Jednou z dominantních výhod tohoto programu je možnost v krátkém časovém úseku simulovat dny i týdny provozu dané sítě v reálném čase. Ovšem rychlost simulace je odvozena především od výkonu počítače, na kterém je prováděna.

OPNET IT Guru je volně ke stažení, ale je nutná registrace, po které je obdržen kód k aktivaci licence nutné pro spuštění.



Obr. 2.2: Úvodní plocha programu OPNET IT Guru

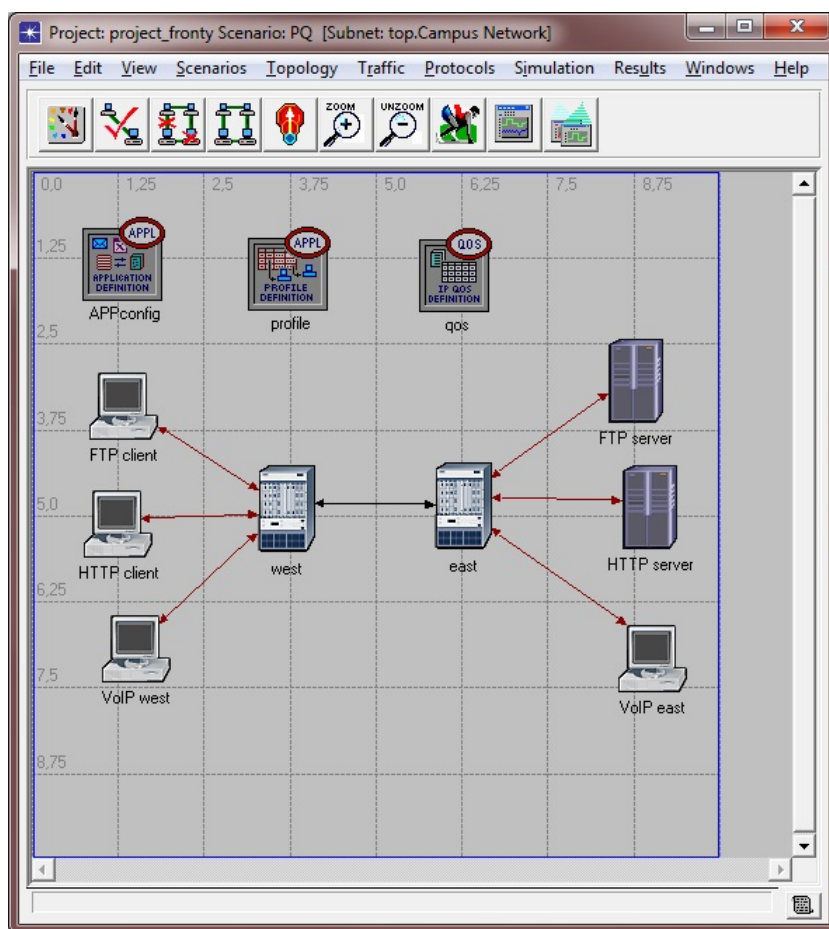
OPNET IT Guru obsahuje v položce *Help* *Tutoriál*, který ukazuje základní rozložení prvků a možnosti porovnání dvou sítí. Dále pak nabízí několik předpřipravených projektů (např. ATM, IP, IP_QoS, Frame_Relay, OSPF, RIP, TCP a mnoho dalších). Tyto projekty obsahují mimo scénáře také návody *ReadMe*, kde se dozvíme základní informace o daných projektech, popis technik a protokolů, pro které jsou vytvořeny, a pomocný návod, jak danou techniku nakonfigurovat.



Obr. 2.3: Ukázka *Helpu* pro předpřipravený projekt IP_QoS

V OPNET IT Guru se pohybujeme podobně jako ve většině jiných programů. Pomocí menu v základní ploše programu lze vytvářet nové, mazat již nechtěné a otevírat již dříve vytvořené projekty. Dále můžeme konfigurovat možnosti samotného programu v položce *Preferences*, ale pokud se nevěnujeme simulacím pomocí OPNET IT Guru profesionálně, tak bychom si mohli spíše ublížit než pomoci.

Po vytvoření či otevření projektu se nám otevře grafický editor, v němž probíhá sestavování sítí a jejich konfigurace.



Obr. 2.4: Ukázka editoru

Při vytváření projektu lze určit, v jaké lokalitě se daná síť nachází. OPNET IT Guru obsahuje velkou škálu slepých map, které tvoří podklad v grafickém editoru. Toho lze využít například při vytváření metropolitních nebo celosvětových sítí. Mapový podklad slouží především pro orientaci v síti a k její lepší přehlednosti.

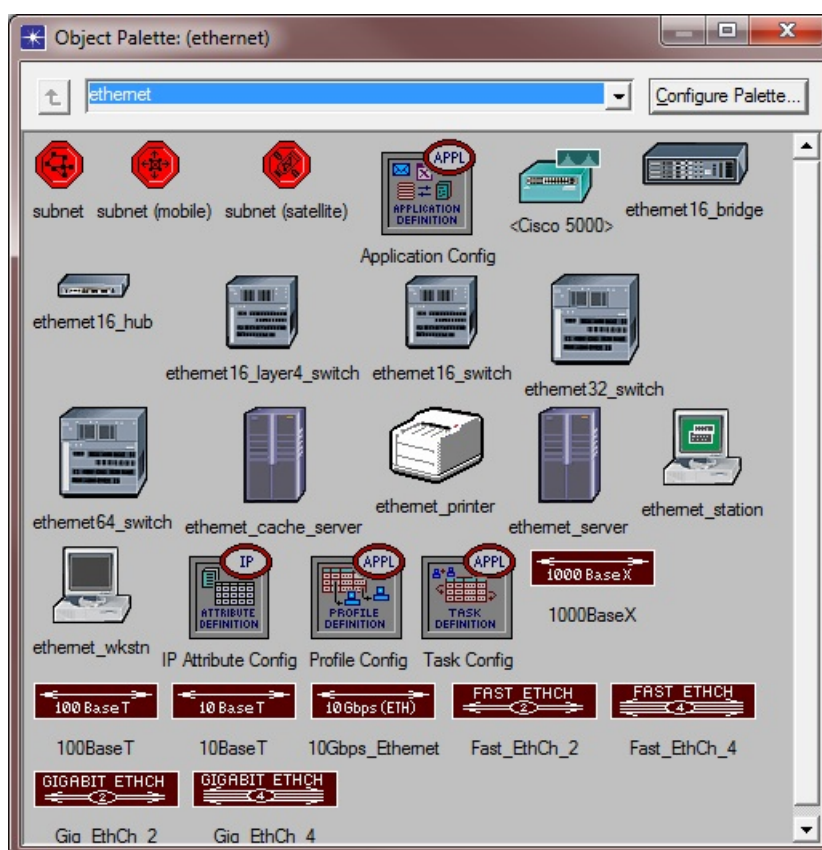
Pomocí editoru lze celou síť „naklikat“ během několika minut, a to díky možnosti kopírování, posunování a uchopování všech komponentů, které si zvolíme.



Obr. 2.5: Hlavní menu projektového editoru a hlavní ikony projektového editoru

K výběru jednotlivých komponentů slouží ikona palety .

Po kliknutí na ikonu palety se otevře dialogové okno, které zprostředkovává výběr mezi všemi knihovnami komponentů, které lze v OPNET IT Guru najít.



Obr. 2.6: Paleta objektů – knihovna komponent pro ethernetové sítě

V programu OPNET IT Guru bylo vypracováno počítačové cvičení (viz B) včetně zadání, postupu vypracování, kontrolních úkolů a požadovaných výsledků pro učitele.

3 ZÁVĚR

Cílem práce bylo proniknout do problematiky různých řešení datových přenosů v integrovaných sítích. Zabezpečení různých tříd kvality služby je jednou z nejdůležitějších oblastí dnešního vývoje datových sítí.

Pro pochopení funkce přenosů dat v síti byly teoreticky rozebrány nejčastěji používané síťové prvky, na jejichž výběru závisí funkčnost sítě.

Dále bylo na základě zadání zjišťováno, zda je možné využít nástrojů MATLAB a SIMULINK pro simulaci činnosti různých síťových prvků či neuronových sítích. Pro neuronové sítě má MATLAB přímo vytvořenou knihovnu **Neural Network Toolbox** s jejíž pomocí lze neuronové sítě vytvářet, trénovat a simulovat.

Pomocí této knihovny a jejích nástrojů bylo vytvořeno počítačové cvičení, které má studentům představit neuronové sítě. Toto téma počítačového cvičení bylo vybráno především z toho důvodu, že do vypracovávání semestrálního projektu a bakalářské práce jsem o neuronových sítích a jejich využití věděl minimum. Dalším důvodem bylo, že většina vrstevníků ani nevěděla, že něco jako neuronové sítě existuje a za dobu bakalářského studia jsme se s podobnou úlohou, která by problematiku neuronových sítí představila nesetkali.

Druhé počítačové cvičení bylo zpracováno na téma modelů kvality služby. V programu OPNET IT Guru byly vytvořeny 2 projekty, které studentům umožní další potřebné konfigurace a možnost porovnat modely IntServ a DiffServ na praktickém příkladě.

Při vytváření laboratorní úlohy v programu OPNET IT Guru byl použit především zdroj [8]

Poslední třetí laboratorní úloha podle dohody s vedoucím práce na téma Simulace síťového prvku pomocí programu MATLAB a Simulink se zaměřuje na simulaci řazení paketů dle priority a požadovaného výstupního portu. Simulace je řízena klasicky, bez použití neuronových sítí.

V této práci byla teoreticky rozebrána problematika síťových přenosů s ohledem na kvalitu služby a problematika základů k neuronovým sítím. Největší přínos padá na vytvořená počítačová cvičení, která mají být použita při výuce studentů.

Zadání, postupy a požadované výsledky pro počítačová cvičení jsou v práci vedeny jako Příloha 1, Příloha 2 a Příloha 3.

Na přiloženém CD jsou jak data nutná pro počítačová cvičení, tak i samotný text bakalářské práce.

LITERATURA

- [1] BLUNÁR, K., DIVIŠ, Z. Telekomunikační sítě. VŠB-TU, Ostrava 2003.
- [2] PUŽMANOVÁ, R. PUŽMANOVÁ, R. *Moderní komunikační sítě od A do Z*. Computer Press, Brno 2006
- [3] HERMAN, Ivo. *Komunikační technologie*. Skripta. Vysoké Učení Technické v Brně
- [4] ŠNOREK, M.: *Neuronové sítě a neuropočítače*. Skripta, Vydavatelství ČVUT, Praha, 2002
- [5] Integrovaný *help* v programu MATLAB
- [6] HOSEK, J. *Nové metody zajištění kvality služeb v datových sítích*, Dizertační práce FEKT, VUT v Brně, 2011
- [7] BRANDEN, R., ZHANG, L., BERSON, S. and JAMIN, J., *Resource Reservation Protocol (RSVP) Version 1 Functional Specification, RFC 2205, September 1997*
- [8] MOLNÁR, K., ZEMAN, O., *Moderní síťové technologie, Laboratorní cvičení* [on-line]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačním technologií, 2006, s.91. Dostupné na: <http://www.utko.feec.vutbr.cz/~molnar/mmos/MMOS_lab.pdf>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AF	zajištěné předávání – Assured Forwarding
ATM	Asynchronous transfer mode
BE	Best Effort
CIR	Committed information rate
DiffServ	diferencované služby – Differentiated Service
EF	urychlené předávání – Expedited Forwarding
FIFO	First In First Out
FTP	File Transfer Protocol
IntServ	integrované služby – Integrated Services
IP	internet protocol
MTU	Maximum Transmission Unit
OSPF	Open Shortest Path First
PQ	Priority Queuing
QoS	kvalita služby – Quality of Service
RIP	Routing Information Protocol
RSVP	Resource reSerVation Protocol
TCP	Transmission Control Protocol
TTL	dobu životnosti – Time To Live
VLAN	Virtual Local Area Network
WFQ	Weighted Fair Queuing

SEZNAM PŘÍLOH

A Příloha 1	46
A.1 Základy k neuronovým sítím	46
B Příloha 2	50
B.1 Konfigurace QoS mechanismu OPNET IT Guru	50
C Příloha 3	59
C.1 Síťové prvky	59

A PŘÍLOHA 1

A.1 Základy k neuronovým sítím

Úvod

Tato úloha byla vytvořena pro první kontakt studenta s pojmem neuronová síť. Jejím úkolem je seznámení studenta s pojmy neuron, neuronová síť a s jejich základním simulováním.

Student by si měl prohlédnout tutoriály, ve kterých je možné vyzkoušet si jakým způsobem je ovlivňována výstupní hodnota na výstupu neuronu (vstupy a jejich synaptické váhy)

Student si v úloze vyzkouší simulaci jednoduchých sítí pro pochopení důvodu využívání neuronových sítí ve výpočetní technice.

Zadání

1. Prostudujte si teoretický úvod.
2. V helpu programu MATLAB si najděte demonstrační úlohy pro:
 - Simple Neuron and Transfer Functions
 - Neuron with Vector Input
 - Perceptrons
3. V těchto demonstračních úlohách si vyzkoušejte základní přenosové funkce, ovlivňování výstupu vstupy a jejich vahami. V případě perceptronů si vyzkoušejte především učení a trénování.
4. Vytvořte pomocí grafického nástroje **nntool** MATLABu:
 - Neuronovou síť, která řeší funkci AND
 - Neuronovou síť, která řeší funkci XOR
5. Vypracujte kontrolní úkoly.

Teoretický úvod

Biologický neuron

Viz kapitola 1.4.1

Umělý neuron a umělé sítě

Viz kapitola 1.4.3 a kapitola 1.4.2

Formální neuron - McCulloh-Pittsův perceptron

Viz kapitola 1.4.4

Neural Network Toolbox

Viz kapitola 2.1.1

Postup

1. Zapněte program MATLAB.
2. Otevřete si *Help* a najděte balík Neural Network Toolbox
3. V části **Demos** si najděte potřebná demo dle zadání 2. a prostudujte si je
4. Vytvořte neuronovou síť ze zadání 4
 - (a) Vytvořte neuronovou síť, která řeší funkci AND.
 - i. Do příkazového řádku napište příkaz **nntool**
 - ii. V novém dialogovém okně (obr. 2.1) Zvolte **New**
 - iii. Přepněte se do záložky **Data**
 - V pravé části vyberte **Inputs**
 - Vhodně pojmenujte potřebná data (např. Inputs)
 - Zadejte vektor pro vstupy funkce AND [0 0 1 1; 0 1 0 1]
 - Klikněte na tlačítko **Create**
 - V pravé části vyberte **Targets**
 - Vhodně pojmenujte potřebná data (např. Targets/Cíl)
 - Zadejte vektor odpovídající výstupu funkce AND [0 0 0 1]
 - Klikněte na tlačítko **Create**

- iv. Přepněte se do záložky **Network**
 - Zadejte vhodné jméno (např. funkce_AND).
 - Typ sítě nastavte **Feed-forward backprop**
 - Input ranges **[0 1; 0 1]** a **Get from input**
 - Training function **TRAINLM**
 - Adaptation learning function **LEARNGDM**
 - Performance function **MSE**
 - Number of layers **2**
 - Layer 1: Number of neurons **2**, Transfer function **TANSIG**
 - Layer 2: Number of neurons **1**, Transfer function **TANSIG**
 - Klikněte na tlačítko **Create**
 - v. Poklepáním na název naší sítě v podokně **Networks** otevřete danou síť. Otevře se Vám nové dialogové okno, ve kterém je znázorněná vytvořená síť.
 - vi. Nyní je potřeba trénování. Přepněte se na záložku **Train**
 - V **Training Info** nastavte **Inputs, Targets** (dle Vámi zadaných názvů).
 - V **Training Parameters** zkontrolujte počet epoch (1000)
 - Klepněte na tlačítko **Train Network**
 - vii. Po trénování sítě ji konečně můžeme odsimulovat
 - V záložce **Simulate** nastavte **Inputs**
 - Výstup simulace se uloží do databáze výsledků
 - Klikněte na tlačítko **Simulate Network**
 - Vraťte se do Neural Network/Data Manageru (nntool)
 - Otevřete si v pravé části vygenerovaný **output** a diskutujte výsledek
- (b) Obdobně vytvořte neuronovou síť, která řeší funkci XOR

Kontrolní úkoly

1. Popište výhody a nevýhody grafického nástroje nntool.
2. Diskutujte, k čemu jsou neuronové sítě dobré v telekomunikační technice.

Závěr

1. Vymažte všechny Vámi vytvořené soubory v MATLABu.
2. Vypněte program MATLAB a počítač

Výsledky měření – pro kontrolu vyučujícím

Výsledkem by měl být správný výstupní vektor. AND [0 0 0 1]. XOR [0 1 1 0].
Místo 0 je vždy výsledné číslo blízké nule (např. -1,13216e-008 a podobně).

Kontrolní úkoly:

1. Popište výhody a nevýhody grafického nástroje nntool.
 - Výhody:
 - Jednoduchost,
 - přehlednost,
 - rychlost práce,
 - možnost exportu/importu.
 - Nevýhody:
 - Neschopnost ovlivnit řadu věcí
 - Pro plnohodnotnou práci na vyšší úrovni je vhodné pracovat přímo ve workspace, kde ovlivníme odkud budeme načítat data, kam se budou ukládat a můžeme sledovat simulaci v mezikrocích.
2. Diskutujte, k čemu jsou neuronové sítě dobré v telekomunikační technice.
 - Neuronové sítě se využívají například pro:
 - Klasifikaci
 - Predikci
 - Řízení
 - Kompresi dat

Klasifikace

Využití pro rozpoznávání tříd pro data, textu, obrázků a jiné.

Predikce

Využití pro metody odhadů vývoje časových řad (využití na burzách i k vědeckým účelům).

Řízení

Využití například při řízení aktivních síťových prvků.

Kompresi dat

Využití například při kompresi obrázků.

B PŘÍLOHA 2

B.1 Konfigurace QoS mechanismu OPNET IT Guru

Úvod

Tato úloha byla vytvořena jako nástroj pro zopakování základních znalostí z oblasti QoS a porovnání základních mechanismů s rozšířenějším protokolem.

Student si zopakuje teorii k obecnému chápání QoS a k metodám integrovaných a diferencovaných služeb.

V programu OPNET IT Guru si student vytvoří síť a odsimuluje si v ní mechanismus řazení do front(chápaný jako zástupce modelu DiffServ) a protokol RSVP (Resource reSerVation Protocol)(chápaný jako zástupce IntServ).

Zadání

1. Prostudujte si teoretický úvod.
2. V programu OPNET IT Guru vytvořte síť pro simulace mechanismu řazení do front(FIFO, PQ, WFQ) a síť pro protokol RSVP.
3. Odsimulujte dané síť a vypracujte kontrolní úkoly.

Teoretický úvod

QoS obecně

Viz kapitola 1.2

IntServ

Viz kapitola 1.2.1

DiffServ

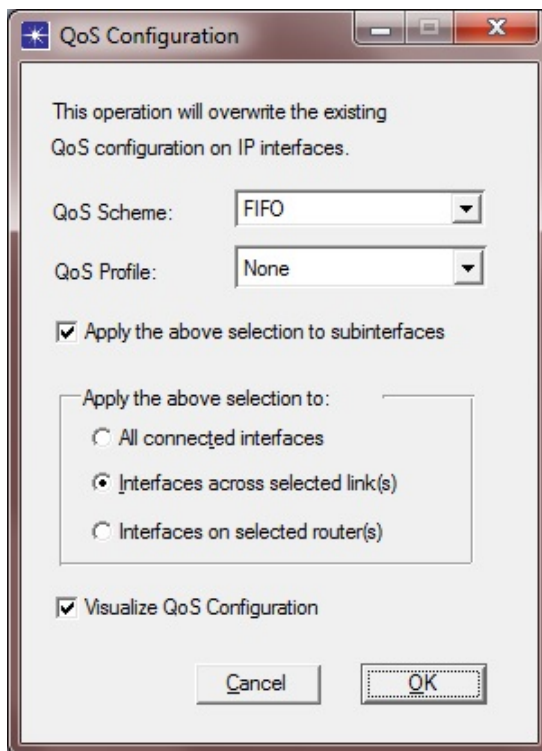
Viz kapitola 1.2.2

Postup

Jelikož nastavování všech potřebných konfigurací zabere velké množství času, byl pro Vás vytvořen scénář obsahující nastavení mechanismu FIFO, se kterým budeme dále pracovat a scénář obsahující konfiguraci RSVP protokolu. Důležité pro nás bude sledování počtu zahozených paketů a koncové zpoždění paketů u služby Voice pro dané mechanismy. U RSVP protokolu si ověříme funkci zpráv pro vytvoření rezervace síťových prostředků.

Mechanismy front

1. Zapněte program OPNET IT Guru Academic Edition 9.1
2. V položce **File** vyberte **Open**
3. Najděte projekt pod názvem **Mechanismy__front**. Otevře se nám projekt a scénář **FIFO**.
4. Nyní potřebujeme nastavit mechanismus FIFO:
 - Klikneme na černou linku mezi směrovači 1 a 2
 - Z hlavní nabídky vybereme **Protocols–IP–QoS–Configure QoS**
 - V dialogovém okně zkontrolujeme nastavení dle obrázku B.1



Obr. B.1: Nastavení FIFO

5. Nyní vybereme sledované statistiky:
 - Pravým kliknutím kdekoli v ploše otevřeme nabídku, vybereme **Choose individual statistics** a zkontrolujeme nastavení:
 - **Global Statistics-IP** :
 - Traffic Dropped (Packets/sec)
 - **Global Statistics-Voice** :
 - Packet end-to-end Delay (sec)
6. Nyní vytvoříme scénáře pro PQ a WFQ
 - V menu **Scenarios** vyberte **Duplicate Scenario** a pojmenujte PQ
 - Nyní potřebujeme nastavit mechanismus PQ:
 - Klikneme na černou linku mezi směrovači 1 a 2
 - Z hlavní nabídky vybereme **Protocols-IP-QoS-Configure QoS**
 - V dialogovém okně zkontrolujeme **QoS Scheme: Priority Queuing** a **QoS Profile: ToS Based**
 - V menu **Scenarios** vyberte **Duplicate Scenario** a pojmenujte WFQ
 - Nyní potřebujeme nastavit mechanismus WFQ:
 - Klikneme na černou linku mezi směrovači 1 a 2
 - Z hlavní nabídky vybereme **Protocols-IP-QoS-Configure QoS**
 - V dialogovém okně zkontrolujeme **QoS Scheme: WFQ** a **QoS Profile: ToS Based**
7. Projekt uložte!
8. Nyní přejdeme k simulaci:
 - V menu zvolte **Scenarios-Manage Scenarios**
 - V kolonce **Results** je potřebné změnit hodnoty na **Collect**, **Recollect**
 - Stiskněte tlačítko **OK**
 - Vyčkejte do konce simulace a poté klikněte na tlačítko **Close**
9. Zvolte **Results-Compare Results**
10. Zobrazte si výsledné statistiky a vysvětlete co znamenají. (Volete zobrazení všech scénářů v jednom grafu)

RSVP

1. V položce **File** vyberte **Open**
2. Najděte projekt pod názvem **prokekt_RSVP1**. Otevře se nám projekt a scénář **no_RSVP**.
3. V tomto scénáři je nastavená jednoduchá síť, kde **klient1** představuje vysílač a **klient2** přijímač dat třídy voice.
4. Tento scénář slouží pro přenos dat bez uvažování služby QoS.
5. Nyní vytvořte scénář pro uvažování QoS.
 - V menu **Scenarios** vyberte **Duplicate Scenario** a pojmenujte **RSVP**
6. Přidejte objekt QoS (pomocí ikony **Paleta**). Zde je potřeba nastavit položky:
 - **RSVP Flow Specification**
 - Počet row nastavte 2
 - * row 0: name: RSVP_IN
 - * row 1: name: RSVP_OUT
 - **RSVP Profiles**
 - Počet row nastavte 2
 - * row 0:
 - (a) Name: RSVP_profil
 - (b) Rerty Policy:
 - i. Retry: retry
 - ii. Retry Timer: 10
 - iii. Reservation Parameters:
 - Flow Description: RSVP_IN
 - * row 1:
 - (a) Name: RSVP_OUT_profil
 - (b) Rerty Policy:
 - i. Retry: retry
 - ii. Retry Timer: 10
 - iii. Reservation Parameters:
 - Flow Description: RSVP_OUT
7. Sledované charakteristiky máme nastavené již od začátku, takže můžeme spustit simulaci obou scénářů:
 - V menu zvolte **Scenarios-Manage Scenarios**
 - V kolonce **Results** je potřebné změnit hodnoty na **Collect**, **Recollect**
 - Stiskněte tlačítko **OK**
 - Vyčkejte do konce simulace a poté klikněte na tlačítko **Close**

8. Zvolte **Results-Compare Results**
9. Zobrazte si výsledné statistiky a vysvětlete co znamenají. (Volte zobrazení obou scénářů v jednom grafu)

Kontrolní úkoly

1. Vysvětlete rozdíl mezi IntServ a DiffServ na nějakém příkladu.
2. Kdy je výhodné použití mechanismu FI-FO (First In-First Out)?
3. Kdy je výhodné použití mechanismu WFQ (Weighted Fair Queuing)?
4. Vysvětlete rozdíl mezi PQ (Priority Queuing) a WFQ.
5. Popište zjednodušeně protokol RSVP (Resource reSerVation Protocol).

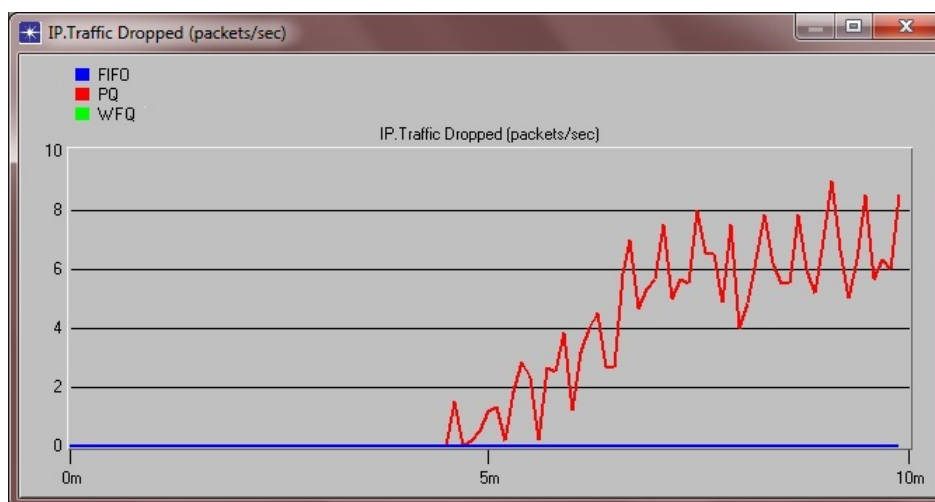
Závěr

1. Vymažte všechny Vámi vytvořené soubory v OPNET IT Guru.
2. Vypněte program OPNET IT Guru a počítač

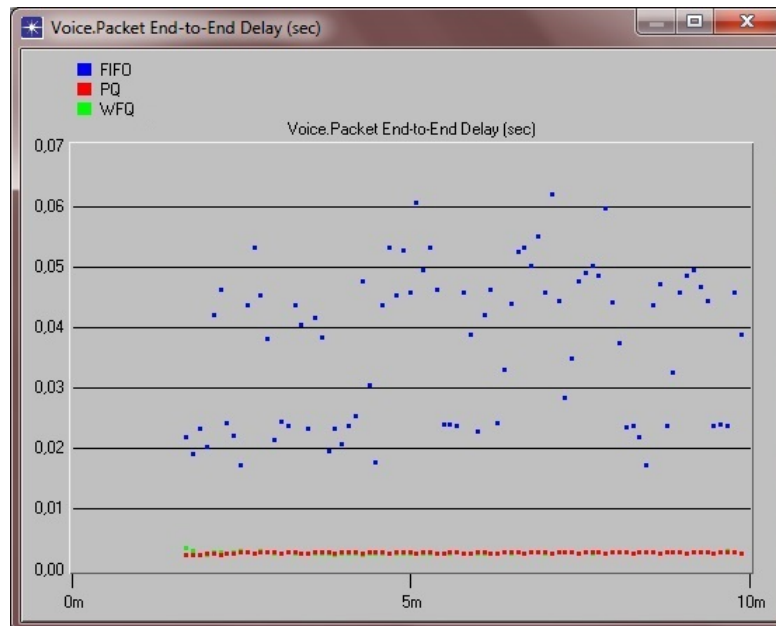
Výsledky měření – pro kontrolu vyučujícím

Mechanismy front

Přibližné grafy z požadovaných charakteristik:



Obr. B.2: Graf zobrazující množství zahozených paketů pro jednotlivé mechanismy



Obr. B.3: Graf zobrazující zpoždění paketů služby Voice pro jednotlivé mechanismy

Graf B.2 ukazuje celkové množství zahozených paketů při simulaci:

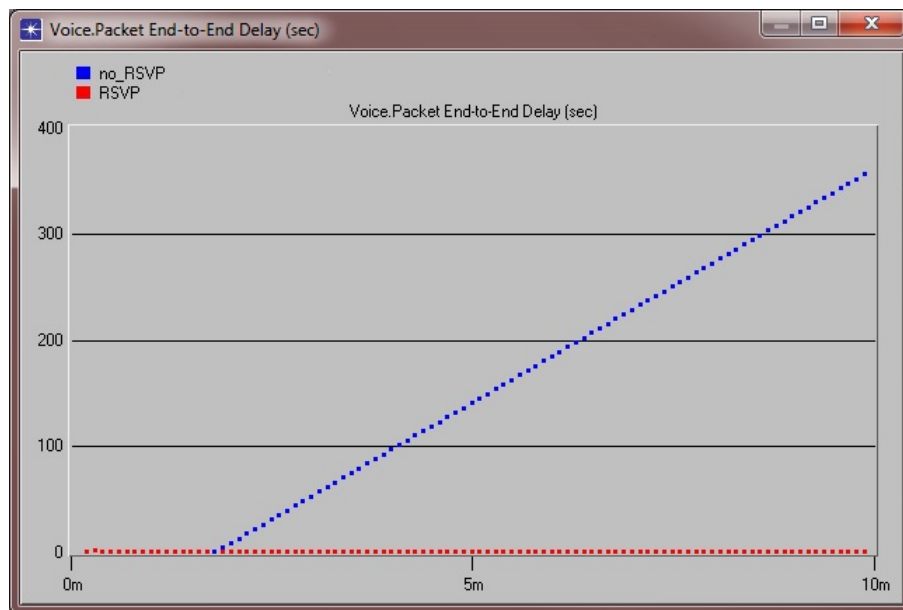
- FIFO žádné zahození: mechanismus v simulaci stíhá posílat pakety dostatečně rychle
- PQ zahazování: mechanismus neustále kontroluje, zda není ve frontě s nejvyšší prioritou nějaký provoz a ve chvíli příchozího paketu stopne provoz na nižších frontách. Tím se zpomaluje a dochází k zahazování paketů nižších tříd, protože se tyto fronty zastavují kvůli jakémukoliv provozu na nejvyšší třídě a samy se zaplňují až dojde k úplnému zaplnění a zahazování paketů.
- WFQ žádné zahození: spravedlivým hospodařením mezi frontami nedochází k zaplnění nižších front a tím k zahazování paketů.

Graf B.3 ukazuje koncové zpoždění paketů služby Voice pro dané mechanismy.

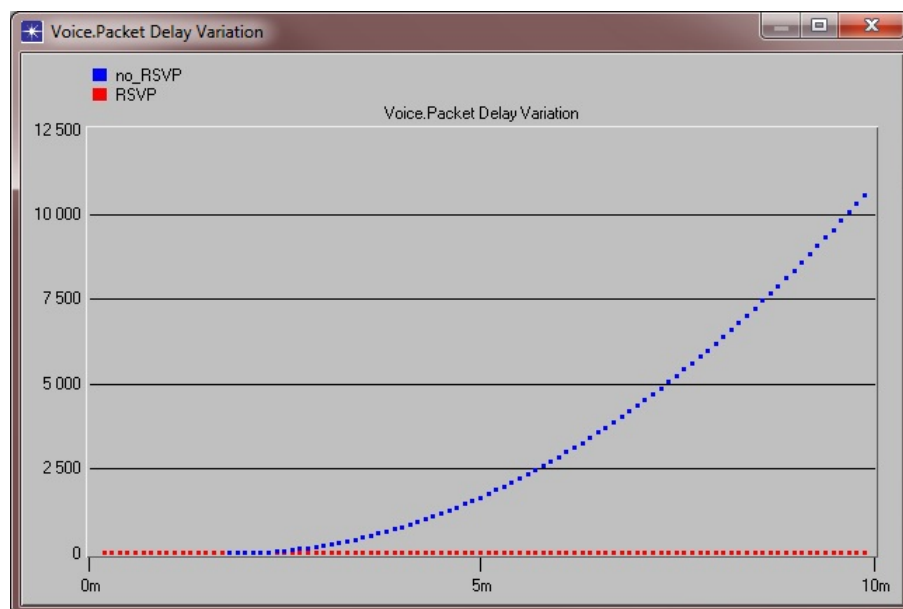
- FIFO zpoždění: řazením všech paketů za sebe bez ohledu na třídu dat dochází k roztroušení paketů služby Voice mezi pakety FTP a HTTP a tím jejich přenos zpožďuje.
- PQ minimální zpoždění: mechanismus striktně dává přednost v přenosu paketům služby Voice.
- WFQ minimální zpoždění: mechanismus stíhá odesílat pakety dostatečně rychle. V případě mnohem většího vytížení ze strany FTP a HTTP by mohlo vzhledem ke spravedlivému zacházení se všemi frontami k jistým zpožděním.

RSVP

Přibližně správné porovnání výsledků simulací sítí s a bez podpory QoS. U sítě bez podpory dochází ke koncovému zpoždění a k jeho kolísání.



Obr. B.4: Graf zobrazující koncové zpoždění paketů pro síť s a bez podpory QoS



Obr. B.5: Graf zobrazující kolísání koncového zpoždění paketů pro síť s a bez podpory QoS

Kontrolní úkoly

1. Vysvětlete rozdíl mezi IntServ a DiffServ na nějakém příkladu.

Například:

- 5MB linka, po které chceme najednou 10 1MB komunikací typu voice.
 - Služba IntServ založená na rezervaci síťových prostředků (šířka pásma) dokáže rezervovat prostředky pouze pro 5 komunikací, zbytek nemůže proběhnout.
 - Služba DiffServ založená na třídění provozu dle důležitosti je schopná s jistou chybovostí zajistit průběh všech 10 komunikací.
2. Kdy je výhodné použití mechanismu FI-FO (First In-First Out)?
 - V málo vytížených sítích.
 3. Kdy je výhodné použití mechanismu WFQ (Weighted Fair Queuing)?
 - Ve vytížených sítích, kde často probíhají různé typy datových přenosů.
 4. Vysvětlete rozdíl mezi PQ (Priority Queuing) a WFQ.
 - PQ striktně propouští pakety z nejvyšší třídy provozu. Ve chvíli, kdy ve frontě této třídy není žádný provoz, můžou jít na výstup pakety z nižší třídy.
 - Při WFQ je zajištěno „spravedlivé“ odesílání paketů ze všech front. Každý fronta má časový interval, ve kterém může odesílat pakety i ve chvíli, kdy je provoz ve frontě s vyšší třídou.
 5. Popište zjednodušeně protokol RSVP (Resource reSerVation Protocol).
 - Viz kapitola 1.2.1

C PŘÍLOHA 3

C.1 Síťové prvky

Úvod

Tato úloha byla vytvořena pro zopakování znalostí z oblasti síťových prvků. Jejím úkolem je prohloubení znalostí v této problematice.

Studen si v úloze zopakuje práci v prostředí Simulink a sám nastaví různé parametry pro správné směrování s ohledem na QoS.

Zadání

1. Prostudujte si teoretický úvod.
2. V programu MATLAB a prostředí Simulink se seznámte se zapojením generátoru paketů.
3. Po pochopení funkce generátoru nastavte požadované parametry v subsystému nazvaném *Síťový prvek*.
4. Vypracujte kontrolní úkoly.

Teoretický úvod

QoS obecně

Viz kapitola 1.2

Síťové prvky

Viz kapitola 1.3

Postup

Jelikož sestavování celého generátoru paketů by časově vyšlo na samostatnou úlohu, je již sestavený a nastavený. Jedná se o generátor, kde se nastavují 2 atributy a to: adresa a priorita. Nás bude zajímat především blok *Síťový prvek*, kde budeme nastavovat řízení adresace na výstupní porty se zohledněním kvality služeb.

1. Přečtěte si teoretický úvod.
2. Zapněte program **MATLAB** a prostředí **Simulink**.
3. Otevřete soubor s názvem **smerovac.mdl**
 - Otevře se Vám schema celého zařízení.
 - Rozklikněte si subsystemy s názvem **Port 1-3** a prostudujte si zapojení.
4. Rozklikněte si subsystem **Síťový prvek**
 - Rozklikněte subsystem **adresování**.
 - V entitě **Output Switch** nastavte:
 - * Switching criterion: **From attribute**
 - * Attribute name: **Adresa**
 - * Tím zajistíme, aby na jednotlivé výstupy odcházely pakety s danou adresou(1-3).
 - Na každý z výstupů přidejte za entitu **FIFO Queue** entitu **Get Attribute**. Na výstup **A1** připojte **Scope**(pomocí osciloskopu můžeme kontrolovat správnost směrování).
 - Na výstup **OUT** připojte původní **Connection bloky**.
 - Dalším bodem bude nastavení subsystemu priority:
 - Rozklikněte subsystem **priority**.
 - V entitě **Output Switch** nastavte:
 - * Number of entity output ports: **5**
 - * Switching criterion: **From attribute**
 - * Attribute name: **Priorita**
 - Na každý z výstupů připojte entitu **FIFO Queue** a za ni můžete přidat již známou entitu **Get Attribute(A1-Scope)**, pomocí které můžeme sledovat správnost řazení paketů dle priority.
 - Nyní najděte vhodnou entitu, kterou spojíte jednotlivé výstupy do jednoho datového toku.
 - Subsystem **priority** přidejte na každý ze tří výstupů subsystemu adresace.

5. Vraťte se do základního okna schématu(**smerovac.mdl**)
 - Na každý výstup ze subsystému **Síťový prvek** připojte entitu **Get Attribute**.
 - V něm vytvořte sledované atributy:
 - * **Adresa**
 - * **Priorita**
 - Na výstup **OUT** připojte **Entity Sink** (to je entita, která zajišťuje ukončení datového toku)
 - Na výstupy **A1** a **A2** připojte blok **Out** (pomocí bloků **Out** se nám do tabulky exportují data ze zvolených atribut)
6. Uložte projekt.
7. Spusťte simulaci.
8. Pokud jste přidávali prvky **Get Attribute** a na jejich výstupy **Scope**, tak si je rozklikněte. Osciloskopy Vám zobrazí hodnoty sledovaných atributů.
9. Nyní je čas na kontrolu výsledků simulace.
 - Ve **workspace** MATLABu se Vám vytvořila tabulka s názvem **yout** (do ní se zapisovaly data z bloků **Out**, které jsme přidávali).
 - Pokud jste neměnili nastavení generátorů ve vstupních portech, tak prvních 6 sloupců znázorňuje data před vstupem do Síťového prvku ve formátu: **Priorita-Adresa**
 - Prostudujte si zbylých 6 sloupců (opět formát **Priorita-Adresa**)

Kontrolní úkoly

1. Jakým způsobem se v generátoru paketů zapisují jednotlivé atributy (adresa a priorita)?
2. Pomocí čeho dostáváme vyexportovaná data do Workspace?
3. Na jaké vrstvě pracují směrovače?
4. Proč ve vyexportovaných datech nenacházíme některé pakety?
5. Entita Switch(input nebo output): Jaký je rozdíl mezi kritériem From attribute a round robin? Co Vám round robin připomíná (obsluha front).

Závěr

1. Pomocí příkazu **clear** v Command Window vymažte vyexportovaná data.
2. Vypněte program MATLAB a počítač.

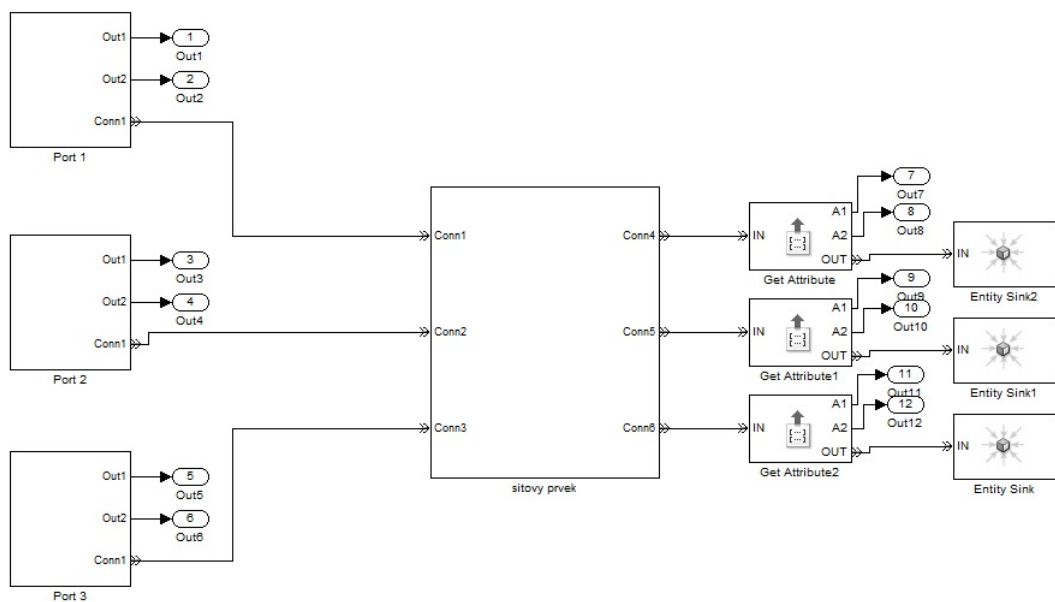
Výsledky měření – pro kontrolu vyučujícím

Výsledkem úlohy je správné zapojení, nastavení požadovaných atribut a vypracování kontrolních úkolů.

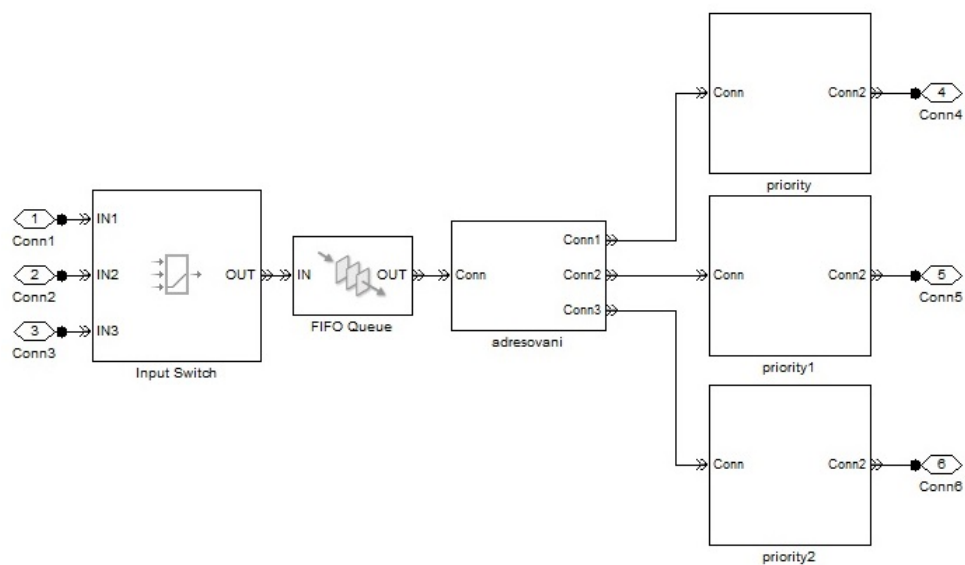
Kontrolní úkoly

1. Jakým způsobem se v generátoru paketů zapisují jednotlivé atributy (adresa a priorita)?
 - Pomocí entity Set Attribute. Vytváříme si atributy, kterým přiřazujeme požadované hodnoty
2. Pomocí čeho dostáváme vyexportovaná data do Workspace?
 - Pomocí bloků Out. Data se nám vypisují ve sloupcích v tabulce yout.
3. Na jaké vrstvě pracují směrovače?
4. Proč ve vyexportovaných datech nenacházíme některé pakety?
 - Důvodem je, že při zpracovávání v bloku adresace máme nastavené velikosti front. Při jejich přeplnění se některé pakety zahodí.
5. Entita Switch(input nebo output): Jaký je rozdíl mezi kritériem From attribute a round robin? Co Vám round robin připomíná (obsluha front).
 - From attribute: Pracuje tak, že na jednotlivé výstupy posílá pakety pouze s daným atributem.
 - Round robin: První paket přicházející do switchu je poslán na OUT1. Druhá na OUT2 a podobně, až do posledního výstupního portu. Následně se pokračuje opět na OUT1.
 - Podobné chování jako WFQ (Weighted Fair Queuing) s tím, že časový interval je nastavený na jeden vstupní paket.

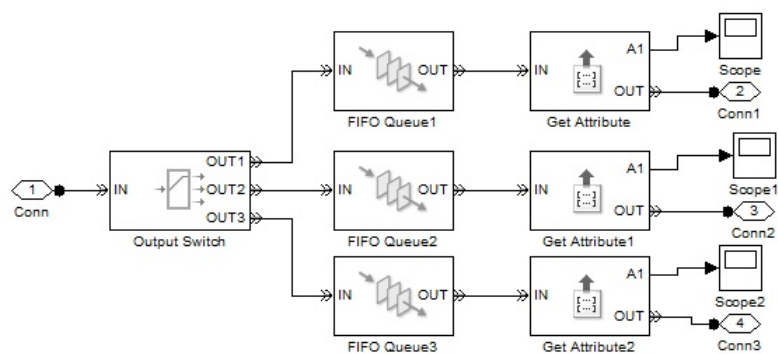
Kontrola zapojení:



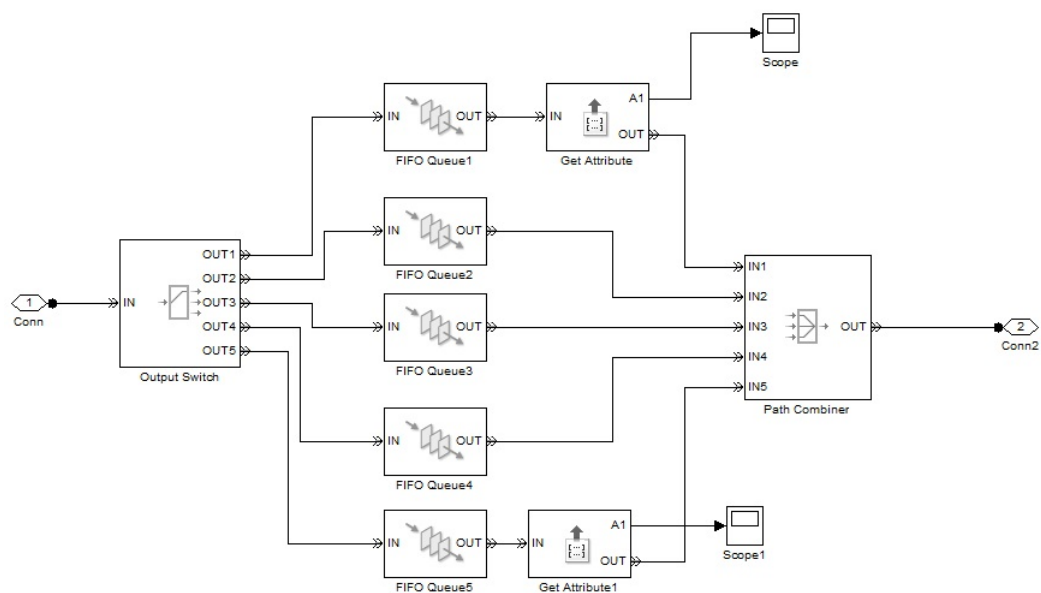
Obr. C.1: Celkové schéma zapojení



Obr. C.2: Subsystem sitový prvek



Obr. C.3: Subsystem adresace



Obr. C.4: Subsystem priority